The topic prevalence between these two groups is visualized to learn about the different themes discussed by the users in these socioeconomic groups. As observed in Fig. 6 and Fig. 7, the difference in topics between the high and low socioeconomic groups differ. The high group has mainly discussed about getting vaccinated, being happy, Instagram, private hospitals, and being secure, while the low socioeconomic groups have mainly discussed about the medical urge, the need for oxygen, pleading with the ministers for financial support, and requesting the Government for more resources, and also free drive for COVID-19 vaccination centers. Even though some of the socioeconomic groups might have a high number of COVID-19 cases due to the high economy and facilities, most of the users overcame the pandemic issues.

For the low socioeconomic groups, the resources are minor, and pleading for oxygen and financial support was the most discussed topic during the second wave of COVID-19.

V. CONCLUSION

In this work, we explore the relationship between social media data, specifically the tweets related to COVID-19 and their relationship with sociodemographic factors, during the second wave of COVID-19. We observe that text analysis, such as sentiment analysis, reveals that the Twitter users tweeted negatively in May 2021, and the number of tweets declined in June 2021, as the second wave has ended. The spatial time pattern analysis revealed that most of the tweets were generated from highly affected COVID-19 states during the second wave, such as Maharashtra, Karnataka, and Kerala, and fewer number tweets from the Northeastern states. The content analysis of Topic modeling has observed that the topics obtained in April, May, and June are grossly different. The sociodemographic factors, such as Population density, Unemployment rate, and GDP of Indian states, are considered and fitted to six supervised classification techniques. The HRFILM has reported the highest accuracy of 96.88% for classifying the socioeconomic groups. The topic prevalence between the sociodemographic groups has suggested that the topics discussed in both groups during the second wave of COVID-19 are different. While the study helps understand the concerns in low socioeconomic groups and can be used by the Government to help them with the needs that they express on social media and for better policy making in the future.

VI. AUTHOR’S CONTRIBUTION

SSSS, SB, RDS have conceptualized the work. SSSS has studied the background, implemented, and reported the results. SB and RDS has supervised the work. All the authors have edited the manuscript.

REFERENCES

Graph Neural Networks for Interpretable Tactile Sensing

Wen Fan  
Dept. of Engineer Mathematics  
University of Bristol  
Bristol, UK  
fanwen2021@gmail.com

Hongbo Bo  
Dept. of Computer Science  
University of Bristol  
Bristol, UK  
hongbo.bo@bristol.ac.uk

Yijiong Lin  
Dept. of Engineer Mathematics  
University of Bristol  
Bristol, UK  
yijiong.lin@bristol.ac.uk

Yifan Xing  
Dept. of Computer Science  
University of Bristol  
Bristol, UK  
yifan.xing@bristol.ac.uk

Weiru Liu  
Dept. of Engineer Mathematics  
University of Bristol  
Bristol, UK  
weiru.liu@bristol.ac.uk

Nathan Lepora  
Dept. of Engineer Mathematics  
University of Bristol  
Bristol, UK  
n.lepora@bristol.ac.uk

Dandan Zhang*  
Dept. of Engineer Mathematics  
University of Bristol  
Bristol, UK  
ye21623@bristol.ac.uk

Abstract—Fine-grained tactile perception of objects is significant for robots to explore the unstructured environment. Recent years have seen the success of Convolutional Neural Networks (CNNs)-based methods for tactile perception using high-resolution optical tactile sensors. However, CNNs-based approaches may not be efficient for processing tactile image data and have limited interpretability. To this end, we propose a Graph Neural Network (GNN)-based approach for tactile recognition using a soft biomimetic optical tactile sensor. The obtained tactile images can be transformed into graphs, while GNN can be used to analyse the implicit tactile information among the tactile graphs. The experimental results indicate that with the proposed GNN-based method, the maximum tactile recognition accuracy can reach 99.53%. In addition, Gradient-weighted Class Activation Mapping (Grad-CAM) and Unsigned Grad-CAM (UGrad-CAM) methods are used for visual explanations of the models. Compared to traditional CNNs, we demonstrated that the generated features of the GNN-based model are more intuitive and interpretable.

Index Terms—Tactile Sensor, Object Recognition, Graph Convolutional Network, Explainability.

I. INTRODUCTION

Vision is the major modality for robotic perception, which can obtain global observation of unstructured environments for robots. However, vision-based object recognition may become challenging due to view occlusions or poor lighting conditions. In this case, tactile perception becomes significant since it can provide robots with an alternative exploration mechanism beyond vision. Therefore, we aim to study tactile-based object recognition in this paper.

Among the existing tactile sensors [1], optical tactile sensors have relatively higher spatial resolution [2]. Light conductive plates, reflective membranes, and displaceable markers have been used to construct optical tactile sensors. Among these mechanisms, the marker displacement-based tactile sensors are easy-to-make, since they can be employed in the arbitrary shape of sensor skin and do not require special lighting arrangement. Therefore, we focus on marker displacement-based optical tactile sensors in this paper. TacTip (see Fig. 1(a) and (b)), developed by Bristol Robotics Laboratory (BRL) [3], [4], is such an example that will be used for experimental studies in this paper.

Recent years have seen promising results of combining deep neural network models with tactile perception. For example, Convolutional Neural Networks (CNNs) have been used for texture classification based on a tactile array sensor [5]. CNNs with Long-Short-Term Memory (LSTM) have been integrated into a whole network as CNN-LSTM and applied to a highly-dense optical tactile sensor GelSight [6] for tactile identification of textures. As for TacTip, CNNs have been used for object classification, edge perception, contour following [7], slip detection [8]. However, most of the traditional CNNs trained for object recognition have millions of parameters [9]. For marker displacement-based tactile sensors which have relatively low spatial resolution, CNNs are not efficient for feature extraction. Therefore, more adequate neural network architectures should be explored.

Fig. 1. As seen in (a) and (b), the TacTip’s skin will deform when interacting with objects. The embedded camera can capture the pins’ movements as tactile information. The tactile graph data can be obtained and analysed based on tactile images, as shown in (c) and (d).
Recently, Graph Neural Network (GNN) has emerged as an alternative to process irregular data, and has demonstrated better performance compared to CNNs [10]. Considering that the pins (also known as markers) of the TacTip can form a graph [4], while the displacements of pins contain rich information that reveals the contacted objects’ shape, we aim to leverage graph-like representations of tactile images for object recognition.

Graph Convolutional Network (GCN) [11], GraphSAGE [12] and Graph Attention Networks (GATs) [13] are representative GNNs. GCN is developed based on applying convolution operation to topological graph, and has been proved to be effective for Physics, Chemistry [14] and social network applications [15], [16]. In this work, we employ GCN-based architecture for tactile object recognition. We aim to compare GCN-based architecture with CNN-based architecture in terms of recognition accuracy and model training speed. Moreover, to ensure that the tactile recognition process is interpretable, Gradient-weighted Class Activation Mapping (Grad-CAM) and Unsigned Grad-CAM (UGrad-CAM) methods are used to enable interpretable tactile sensing [17] [18].

The key contributions of our work include:
- Transform the tactile image data obtained by the TacTip sensor into graph representation;
- Develop an optimal GCN-based model for object recognition based on empirical studies;
- Evaluate the interpretability of the GCN-based model for tactile object recognition.

Applying GNN-based methods to vision-based object classification tasks is popular in deep learning. However, to the best of our knowledge, GNN-based tactile-oriented object recognition with biomimetic optical tactile sensors has been scarcely investigated. Intuitive and accurate tactile perception is a prerequisite for manipulation tasks. Based on our results, GNN-based methods can be more efficient for processing tactile images, and have high potentials to benefit tactile robotics research and can ensure interpretability.

We organize the rest of this paper as follows. Firstly, the tactile graph construction and the architecture of the proposed GCN-based frameworks are introduced in Section II. Secondly, the experiment design and results analysis are described in Section III. Finally, conclusions are drawn in Section IV.

II. METHODOLOGY

A. Hardware Deployment

Ten different 3D-printed objects are used for data collection, as shown in Fig. 2(a). A low-cost desktop robot arm (Dobot Magician) is used for experiments (see Fig. 2(b)). A TacTip sensor is mounted at the wrist of the robot. The optical tactile sensor TacTip used for data collection comprises a 3D-printed soft rubber-like hemispherical skin, whose papillae pins are distributed uniformly on the inner surface of the skin. The displacements of the papillae pins in the inner surface of the skin can be captured by an RGB camera (ELP 1080p module) to generate tactile data.

During data collection, a remote controller was used to guide the robot arm with tactile sensor on its wrist to contact the target object’s surface, while the tactile image data was recorded simultaneously. The whole dataset for 10 class objects are divided into training set (70%), validation set (20%) and test set (10%) for model training and evaluation.

B. Tactile Graph Construction

As shown in Fig. 1(b), 169 pins of TacTip are distributed uniformly in the shape of concentric circles with increasing radius. Among all the pins, the distances between each pair of adjacent pins are almost identical. The graph representations of tactile images are the inputs for our proposed GCN-based framework. The key features of contact deformation on TacTip sensor will be extracted by the proposed model. All the variables used for graph construction are summarised in Table I.

A graph consists of two mandatory components: Nodes (Vertices) and Edges, denoted as $G = (V, E)$. When fed into GNN model, a graph is represented as $G = (X,A)$ where $X$ indicates the node features and $A$ presents an adjacency matrix generated from edges $E$. The white pins of TacTip can be regarded as graph nodes $V$, while the positions of the pins in the tactile images are used as node features $X = \{ v^i = (v^i_x, v^i_y) , i = 1,2,...169 \}$. Let $m$ indicate the total number of edges in one graph. Then undirected edges $E = \{ e^j = [s, t]^T , s \neq t , j = 1,2,...m \}$ can be built between every pair of possible nodes, where $s$ and $t$ represent the source index and target index respectively. As illustrated in Fig. 3, the graph construction process consists of two steps, i.e. i) node extraction, ii) edge connection.

Every raw tactile image obtained by TacTip sensor is cropped and resized to $280 \times 280$, followed by denoising and binarisation (see Fig. 3(a), (b), (c)). Subsequently, the...
blob extraction algorithm, supplied from the OpenCV library, is used to extract the position of each pin as corresponding node features (see Fig. 3(d)). Finally, the k-Nearest Neighbors (kNN), is used to build edge connections (see Fig. 3(e)).

The quality of generated graph is measured in terms of efficiency, connectivity and robustness. We define that the efficiency is high if the graph construction frequency (the number of graphs generated per second) is higher than 50 Hz. We evaluate graph’s connectivity in terms of the neighbors’ number linked to every node. High connectivity requires that each node should be connected with at least four adjacent nodes, while redundant connectivity means connection with six adjacent nodes. The robustness is measured based on the difference between the graph connectivity before and after TacTip sensor interacting with objects.

The numbers of nearest neighbors (k value) selected for the kNN classifier have significant impact on the quality of graph construction. According to the characteristics of the TacTip pins’ distribution, the outmost nodes should have less neighbors than the inner ones. So we explore an adaptive kNN approach to build edge connections, as shown in Fig. 4.

Traditional kNN has a single k value (Fig. 4(a)-(d)), while for our adaptive one, two different parameters $k_1$ and $k_2$ are used to cluster nodes in central and non-central areas of the graph respectively (Fig. 4(e)-(f)). The performances of kNN with different k values are summarized in Table II. Adaptive KNN approach can minimize the generation of redundant connections between outmost nodes and remote neighbors. However, the computation time required for the adaptive kNN approach increases significantly, which leads to low efficiency. According to the results, $k = 6$ is selected as the default value for graph construction, which can ensure desired performance in terms of efficiency, robustness and connectivity.

### Table II

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Edges</th>
<th>Efficiency</th>
<th>Robustness</th>
<th>Connectivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k = 1$</td>
<td>(2, 169)</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>$k = 2$</td>
<td>(2, 338)</td>
<td>High</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>$k = 3$</td>
<td>(2, 507)</td>
<td>High</td>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>$k = 4$</td>
<td>(2, 676)</td>
<td>High</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>$k = 5$</td>
<td>(2, 845)</td>
<td>High</td>
<td>Medium</td>
<td>Redundant</td>
</tr>
<tr>
<td>$k = 6$</td>
<td>(2, 1014)</td>
<td>High</td>
<td>High</td>
<td>Redundant</td>
</tr>
<tr>
<td>$k_1, k_2 = 6, 5$</td>
<td>(2, 971)</td>
<td>Low</td>
<td>High</td>
<td>Redundant</td>
</tr>
<tr>
<td>$k_1, k_2 = 6, 4$</td>
<td>(2, 928)</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>$k_1, k_2 = 6, 3$</td>
<td>(2, 885)</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>$k_1, k_2 = 6, 2$</td>
<td>(2, 842)</td>
<td>Low</td>
<td>Medium</td>
<td>High</td>
</tr>
<tr>
<td>$k_1, k_2 = 6, 1$</td>
<td>(2, 799)</td>
<td>Low</td>
<td>Low</td>
<td>Medium</td>
</tr>
</tbody>
</table>

C. Tactile GNN Framework

The architecture of Tactile GNN model used for object recognition in our paper is shown in Fig. 5. It consists of multiple GCN layers [11], followed by fully-connected layers. A GCN layer can be defined as $\tilde{H} = GCN(H, A) = \sigma(\tilde{D}^{-0.5} \tilde{A} \tilde{D}^{-0.5} H W) = \sigma(V H W)$, where $\tilde{A}$ is the normalized adjacency matrix generated from $A$, $\tilde{D}$ is the degree matrix related to $\tilde{A}$, $W$ is the weight matrix of current GCN layer, and $\sigma(.)$ is the activation function. For example, ReLU is used as the activation function in our work. In the first GCN layer, the degree matrix $\tilde{D}$ is calculated as $\tilde{D}_{ii} = \sum_{j=1}^{n} \tilde{A}_{ij}$.
layer, $H = X$, where $X$ represents the node features.

We examine the design spaces to choose the optimal structure for the GCN-based model. The results can be found in Section III-A and Table III. Specifically, we experimented with different numbers of GCN and FC layers, and also two pooling methods (scatter-max or scatter-mean).

D. Graph Explainable Methods

Follow the definition of GCN layer $H' = \sigma(VH)$ and explainable methods for GCN model in work [18], the $k$th graph convolutional feature map $F^l$ at layer $l$ is set as:

$$F^l_k(X, A) = \sigma(VF^{l-1}(X, A)W^l_k)$$  \hspace{1cm} (1)

The global average pooling feature $e$ of node $n$ from layer $L$ (normally, $L$ is the last GCN layer) should be:

$$e_k = \frac{1}{N} \sum_{n=1}^{N} F^L_{k,n}(X, A)$$  \hspace{1cm} (2)

Then the Grad-CAM’s weights $\alpha$ for class $c$ is calculated by (3), where the score $y$ of class $c$ is $y^c = \sum_k \omega_k e_k$.

$$\alpha_{L,c} = \frac{1}{N} \sum_{n=1}^{N} \frac{\partial y^c}{\partial F^L_{k,n}}$$  \hspace{1cm} (3)

Finally, the heat-map $M$ which can visualise the positive contribution of node $n$ for graph $G(X, A)$ is generated by:

$$M^{c,L,n} = ReLU(\sum_k \alpha_{L,c} F^L_{k,n}(X, A))$$  \hspace{1cm} (4)

A new method called UGrad-CAM has been proposed [18], which can show both positive and negative contributions from nodes. We decide to apply both their Grad-CAM and UGrad-CAM explaining tools on our tactile GNN model.

III. EXPERIMENTS, RESULTS AND DISCUSSION

A. Empirical Evaluations

We conduct empirical evaluations to study how the design settings of GCN layers, pooling methods and FC layers influence the performance of Tactile GNN models. We investigate the performance of 4 types of network structures. The first two classes were defined as ‘max, original FC’ and ‘mean, original FC’, the last two classes were ‘max, standard FC’ and ‘mean, standard FC’. The ‘original FC’ meant that each FC layer’s channels would vary with the last layer’s channels. If using $[(a_i, b_i)](i = 1, 2, ..., I)$ to represent FC structure, then $a_i$ and $b_i$ represented the input and output channel amounts for the $i_{th}$ FC layer, $I$ indicated the total number of FC layers and was set as 3 in this paper. $J$ presented the total number of GCN layers, while $c_j$ represented the output channel numbers for $j_{th}$ GCN layer. For ‘original FC’, $a_i = c_{j+1-i}$, $b_i = c_{j-i}$, while $b_3$ was equal to 10. For ‘standard FC’, $a_1 = c_J$, $a_2 = b_1 = 128$, $a_3 = b_2 = 96$, and $b_3 = 10$. Both pooling methods of scatter-max and scatter-mean were tested to explore their impacts. Adam optimizer (learning rate $\alpha = 10^{-3}$) was used for model training with batch size $\beta = 128$, while early stop mechanism was applied.

The training and evaluation results are summarized in Table III. Based on the results, we notice that the tactile GNN with 7 layers GCN and 3 layers FC has an adequate compromise between computation speed and test accuracy. Increasing the number of GCN layer can enhance the performance, however, the cost of low computation speed is not desirable. The detailed analysis is given below:

1) Depth of GCN: The network prediction accuracy increases when GCN becomes deeper. However, after the depth is greater than 7, the improvement of network prediction accuracy slows down while the cost of training increases dramatically.

Considering the structural features of TacTip and the GNN working principles, the feature aggregation from the center to the outermost nodes (or the opposite direction) would require at least 7 steps (see Fig. 6). This could explain why the predictions improve significantly while the layer number is less than 7. However, network models with 8, 9 and 10 GCN layers were prone to be over-smoothing, which could result in the same representation of most nodes. The time required for training one epoch using 6-layer GCN and 7-layer GCN is 17.4s and 24.9s respectively, which do not have significant difference. Compared to 6-layer GCN, the test accuracy of 7-layer GCN increases 0.4%. The test accuracy for 8-layer GCN is lightly better than the one for 7-layer GCN, while the improvement is less than 0.2%. However, the computation time of the 8-layer GCN is nearly two times longer than that of the 7-layer GCN.
2) **Pooling Methods**: Scatter-mean normally performs better than scatter-max. With the same setting of hyperparameters, the training speed of GNNs using scatter-mean are 2% - 5% faster than those using scatter-max. Moreover, GNNs had higher prediction accuracy when using scatter-mean than scatter-max. These differences were evident when Tactile GNN models have 1 or 2 GCN layers. This may be due to the fact that scatter-mean incorporates all node features, which reduces information loss and allows the FC network to learn more useful features.

3) **FC layers**: The structure of FC layers had significant influence on the model performance. The model with standard FC layers showed better performance than the original FC until the depth of GCN layers increased to 7. When the depth was more than 8, the model with standard FC layers had lightly lower test accuracy compared to the one with original FC layers. However, it had the advantage of higher computation efficiency, since fewer neurons were required for model training.

### TABLE III

<table>
<thead>
<tr>
<th>Depth</th>
<th>Class</th>
<th>Train Time (s/epoch)</th>
<th>Test Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>max, original FC</td>
<td>5.2</td>
<td>62.31%</td>
</tr>
<tr>
<td>1</td>
<td>mean, original FC</td>
<td>5.1</td>
<td>79.90%</td>
</tr>
<tr>
<td>1</td>
<td>max, standard FC</td>
<td>5.2</td>
<td>76.60%</td>
</tr>
<tr>
<td>1</td>
<td>mean, standard FC</td>
<td>5.0</td>
<td>89.01%</td>
</tr>
<tr>
<td>2</td>
<td>max, original FC</td>
<td>6.2</td>
<td>86.87%</td>
</tr>
<tr>
<td>2</td>
<td>mean, original FC</td>
<td>6.2</td>
<td>87.97%</td>
</tr>
<tr>
<td>2</td>
<td>max, standard FC</td>
<td>6.0</td>
<td>93.87%</td>
</tr>
<tr>
<td>2</td>
<td>mean, standard FC</td>
<td>6.0</td>
<td>95.73%</td>
</tr>
<tr>
<td>3</td>
<td>max, original FC</td>
<td>7.9</td>
<td>94.52%</td>
</tr>
<tr>
<td>3</td>
<td>mean, original FC</td>
<td>7.9</td>
<td>94.62%</td>
</tr>
<tr>
<td>3</td>
<td>max, standard FC</td>
<td>7.9</td>
<td>97.29%</td>
</tr>
<tr>
<td>3</td>
<td>mean, standard FC</td>
<td>7.8</td>
<td>97.32%</td>
</tr>
<tr>
<td>4</td>
<td>max, original FC</td>
<td>10.2</td>
<td>96.64%</td>
</tr>
<tr>
<td>4</td>
<td>mean, original FC</td>
<td>10.0</td>
<td>96.84%</td>
</tr>
<tr>
<td>4</td>
<td>max, standard FC</td>
<td>10.2</td>
<td>98.41%</td>
</tr>
<tr>
<td>4</td>
<td>mean, standard FC</td>
<td>10.0</td>
<td>97.88%</td>
</tr>
<tr>
<td>5</td>
<td>max, original FC</td>
<td>13.6</td>
<td>98.16%</td>
</tr>
<tr>
<td>5</td>
<td>mean, original FC</td>
<td>13.3</td>
<td>97.95%</td>
</tr>
<tr>
<td>5</td>
<td>max, standard FC</td>
<td>13.5</td>
<td>98.67%</td>
</tr>
<tr>
<td>5</td>
<td>mean, standard FC</td>
<td>13.2</td>
<td>98.56%</td>
</tr>
<tr>
<td>6</td>
<td>max, original FC</td>
<td>18.0</td>
<td>98.48%</td>
</tr>
<tr>
<td>6</td>
<td>mean, original FC</td>
<td>17.5</td>
<td>98.63%</td>
</tr>
<tr>
<td>6</td>
<td>max, standard FC</td>
<td>17.7</td>
<td>98.70%</td>
</tr>
<tr>
<td>6</td>
<td>mean, standard FC</td>
<td>17.4</td>
<td>98.60%</td>
</tr>
<tr>
<td>7</td>
<td>max</td>
<td>26.2</td>
<td>98.97%</td>
</tr>
<tr>
<td>7</td>
<td>mean</td>
<td>24.9</td>
<td>98.99%</td>
</tr>
<tr>
<td>8</td>
<td>max, original FC</td>
<td>42.6</td>
<td>99.03%</td>
</tr>
<tr>
<td>8</td>
<td>mean, original FC</td>
<td>39.8</td>
<td>99.13%</td>
</tr>
<tr>
<td>8</td>
<td>max, standard FC</td>
<td>42.1</td>
<td>99.04%</td>
</tr>
<tr>
<td>8</td>
<td>mean, standard FC</td>
<td>39.8</td>
<td>99.16%</td>
</tr>
<tr>
<td>9</td>
<td>max, original FC</td>
<td>75.7</td>
<td>99.35%</td>
</tr>
<tr>
<td>9</td>
<td>mean, original FC</td>
<td>73.7</td>
<td>99.35%</td>
</tr>
<tr>
<td>9</td>
<td>max, standard FC</td>
<td>74.7</td>
<td>99.13%</td>
</tr>
<tr>
<td>9</td>
<td>mean, standard FC</td>
<td>72.9</td>
<td>99.28%</td>
</tr>
<tr>
<td>10</td>
<td>max, original FC</td>
<td>151.8</td>
<td>99.49%</td>
</tr>
<tr>
<td>10</td>
<td>mean, original FC</td>
<td>150.2</td>
<td>99.53%</td>
</tr>
<tr>
<td>10</td>
<td>max, standard FC</td>
<td>151.5</td>
<td>99.10%</td>
</tr>
<tr>
<td>10</td>
<td>mean, standard FC</td>
<td>149.9</td>
<td>99.26%</td>
</tr>
</tbody>
</table>

![Fig. 7. The visual explanation results using Grad-CAM and UGrad-CAM.](image)

---

### B. Results Analysis

To ensure that the deep learning-based tactile recognition process is interpretable, Grad-CAM and UGrad-CAM are used to provide visual explanations for GCN models [18].

1) **Comparisons for Recognition Accuracy**: CNN networks with standard FC layers and different numbers of convolutional layers were constructed for comparisons. The training speed of CNNs was much lower than GNNs with same number of layers, while CNNs’ prediction accuracy was slightly higher. A 3-layer CNN provides the best performance with test accuracy of 99.8%. However, the training time for one epoch is 129.2s, which is much longer than 3-layer GCN. If increasing the CNN layers to 7, the training time is 314.9s per epoch while the test accuracy is 99.67%. As for the 7-layer GCN, the test accuracy is 24.9s. This indicates that GCN-based models are computationally efficient.

2) **Comparisons for Interpretability**: We then compare the interpretability of CNNs and GNNs for tactile perception. The examples of Grad-CAM and UGrad-CAM based analysis are shown in Fig. 7. The red regions refer to the areas where the attention from model is strong. For Grad-CAM based analysis, the blue regions indicate the areas contribute less to the models’ decision-making process. As for UGrad-CAM, blue areas represent negative contributions to decision-making.

Fig. 7(a) shows examples that CNNs can successfully extract useful features depending on the deformation. However, they may generate the prediction mostly based on the background information (see Fig. 7(b)), which is not reasonable since the background has been denoised and binarised (see Fig. 4(c)). As for GNNs-based model, attention is paid to the contact regions instead of the background areas. Fig. 7(c)
and (d) show the visual explanations of GCNs-based object recognition process using UGrad-CAM and Grad-CAM respectively. Refer to Fig. 4(d)(e), implicit contact information, such as the location and level of deformation can be visualised.

To quantify the interpretability of both types of models, we randomly selected 100 tactile images for analysis. 55% of tactile images failed to be interpreted by CNNs in a reasonable manner, since the attention of CNNs is located at the background, instead of the deformation. According to the results, GNNs can identify the regions where deformation is caused by interaction with the target object.

IV. CONCLUSION

In this paper, a GCN-based model (Tactile GNN) was proposed for tactile object recognition. The soft biomimetic optica sensor TacTip was used to record tactile images. The performance of Tactile GNN models with different structures were investigated, while the highest test accuracy for object recognition could reach 99.53%. Grad-CAM and UGRAD-CAM were used to evaluate the interpretability of the proposed model. Compared with CNNs, the accuracy for object recognition using GNNs is lower, but the training efficiency is improved significantly. This is due to the fact that Tactile GNNs can extract the key contact information effectively, while CNNs based methods need to process the entire tactile images that contain redundant information. Moreover, Tactile GNN could identify the useful information of deformation, and can be used for interpretable tactile sensing.

We hope the empirical studies and discussions in this paper can provide inspiration for researchers who are interested in GNN-based tactile object recognition with interpretability. We envision that the proposed method can be generalised to other types of tactile sensors and benefits robot learning research where efficient tactile perception is significant.

REFERENCES


State Of Charge estimation using Extended Kalman Filter in Electric Vehicles

Azizeh Lotfivand
Engineering and Tech. Dept.
Liverpool John Moores University
Liverpool, Britain
nazlylof@yahoo.com

Dingli Yu
Engineering and Tech. Dept.
Liverpool John Moores University
Liverpool, Britain
D.Yu@ljmu.ac.uk

Barry Gomm
Engineering and Tech. Dept.
Liverpool John Moores University
Liverpool, Britain
J.B.Gomm@ljmu.ac.uk

Abstract—Lithium-ion batteries are in the top priority among other batteries because of being safe, quick charging, and long-life cycle and are widely used in electric vehicles for having precise battery model, it is essential to decide factors like state of health and state of charge. In this paper using LA92 drive cycle experiment data is used for the state of charge estimation algorithm. Firstly, for imitating the lithium-ion battery’s behavior in accurate way, a mathematical model of analogous battery is required. The 2RC branches are included in Thevenin model and Hybrid Pulse Power Characterization (HPPC) test data achieved at 40°C, 25°C, 10°C, 0°C, and -10°C is used to calculate the SOC 3-dimensional curve as a function of SOC and T. Coulomb counting (CC) and extended Kalman Filter method were observed for estimating the state of charge. The results show that EKF is more precise than CC (Coulomb Counting). The mistake of estimation with EKF is less than 1% that shows the reliability of the algorithm and

Keywords; Electric Vehicles (EV); Lithium-ion batteries (Li-Ion); State of Charge (SOC); Extended Kalman Filter (EKF)

I. INTRODUCTION

Recently, environmental concerns like greenhouse gas removal global climate change and the depletion of energy resources lead to use electrical vehicles (EV). Among several types of batteries, Lithium-ion batteries are more preferable for some characteristics like their small size, high energy density, light weight, high output power, high safety and low self-discharge rates [2]. Nonetheless, aging and temperature are two factors that could influence on their operation. Therefore, emphasize on environment temperature will prevent aging, thermal runaway, and physical injury.

Better performance of lithium-ion batteries, their Safety and longevity we require Battery management System (BMS). An effective BMS has 4 main characteristics as following: (a) keeping the battery temperatures between the certain limits (b) precisely estimating batteries’ state of charge (SOC), state of energy (SOE), state of health (SOH), remaining useful life (RUL), (c) equalizes the voltage, charge, and capacity among battery cells (d) operates fault detection, fault forecasting, and fault overcoming and [2]. Estimate the battery state of charge is allied to main duties of BMS. However, accurate SOC estimation is difficult and cannot be measured directly and multiple factors affect the value, so it is difficult to measure SOC when battery is working; Therefore, the SOC must be estimated. The SOC is the proportion of a battery's usable capacity Q(t) to the maximum charge that can be stored and ranges it from 0% to 100%, with intervals of 10%; however, it can be changed as desired. There are several ways for determining SoC, the most simple and common of them is Coulomb counting (CC). Its precision, however, is dependent on the first SOC and sensor precision [2] [3]. In this method all the sensor errors are compounded together and reflected to the results. Another technique is the OCV-SOC curve method, where OCV represents the battery's open circuit voltage. This method converts open circuit voltage to corresponding SOC. For Li-ion batteries, however, the curve will be plain in the midway, making accurate estimation impossible.

Closed loop control methods such as Extended Kalman Filter (EKF) is proposed in this study for solving the shortcomings of CC and OCV-SOC algorithm.

The equivalent electrical circuit consists of two RC model [4] is used in this paper to estimate state of charge (SOC) with the EKF and UKF algorithms. Simulation data are used to compare the EKF method with the CC method. In the following section the equivalent electrical model of battery and its corresponding equations are discussed in detail. In the next section, multiple estimation approaches for state of charge estimation are presented and compared results and conclusion are discussed in the last part.

II. SELECTED BATTERY MODEL AND CORRESPONDING EQUATIONS

Figure 1. Battery Thevenin Model.
The state and measurement equations are as followings:

\[ V_1(t) = \frac{-V_1(t)}{R_1 C_1} + \frac{i(t)}{C_1} \]  
\[ V_2(t) = \frac{-V_2(t)}{R_2 C_2} + \frac{i(t)}{C_2} \]  
\[ \text{SoC} = \frac{\eta(t)}{Q} \]  
\[ V_f(t) = V_{OC}(\text{SoC}(t)) + V_1(t) + V_2(t) + I(t)R_0 \]  

The state vector is \( x = [V_1, V_2, \text{SoC}] \)

The state matrix is \( A \), the input matrix is \( B \), the output matrix is \( C \), and the feedthrough matrix is \( D \).

EKF approach uses a discrete state space model. This is due to the fact that data will be updated after each time step. Closed loop approaches such as EKF is employed to solve these shortcomings. The suggested methodology employs the EKF method to estimate SOC and Terminal voltage.

The Hybrid Pulse Power Characterization (HPPC) test data achieved in 4 temperature from 40°C to -10°C are used to calculate the SOC 3-dimensional curve as a function of SOC and T. In the proposed methodology, figure 2 is resulted by fitting a four-order polynomial to the entirety of the SOC-OCV data with thermal effects on Open Circuit Voltage.

The prominent and forthright method for estimating SOC is the Coulomb counting method [7]. However, there are two problems using these algorithms: the first amount of SoC and sensor noise [8]. SoC estimation will be erroneous if the starting SoC is incorrect. Because the CC method is not a close loop control method, so sensor noise will be added at each time step. Closed loop approaches such as EKF is employed to solve these shortcomings. The suggested methodology employs the EKF method to estimate SOC and Terminal voltage.

The Hybrid Pulse Power Characterization (HPPC) test data achieved in 4 temperature from 40°C to -10°C are used to calculate the SOC 3-dimensional curve as a function of SOC and T. In the proposed methodology, figure 2 is resulted by fitting a four-order polynomial to the entirety of the SOC-OCV data with thermal effects on Open Circuit Voltage.

\[ \text{OCV} = f(\text{SOC}, \text{Temperature}) \]

It can be written as following:

\[ \text{OCV}_{\text{fit}} = p_{00} + p_{10} \text{SOC} + p_{11} \text{SOC} \times T + p_{20} \text{SOC}^2 + p_{11} \text{SOC} \times T + p_{02} T^2 + p_{30} \text{SOC}^3 + p_{21} \text{SOC}^2 \times T + p_{12} \text{SOC} \times T^2 + p_{13} T^3 + p_{03} T^4 \]

Derivative of OCV with respect to SOC is:

\[ \frac{d\text{OCV}_{SOC}}{d\text{SOC}} = 4p_{40} \text{SOC}^3 + 3p_{31} \text{SOC}^2 + T + 3p_{30} \text{SOC}^2 + 2p_{22} \text{SOC} + T^2 + 2p_{21} \text{SOC} + T + 2p_{20} \text{SOC} + p_{13} T^3 + p_{12} T^2 + p_{11} T + p_{10} \]

Derivative of OCV with respect to SOC is:

\[ \frac{d\text{OCV}}{dT} = p_{31} \text{SOC}^3 + 2p_{22} \text{SOC}^2 + T + p_{21} \text{SOC} + T^2 + 2p_{20} \text{SOC} + p_{13} T^3 + p_{12} T^2 + p_{11} T + p_{10} \]

These derivatives are required for C matrix to linearization and achieving Jacobians.
According to figure 2 it can be seen that the 3-dimensional curve has acceptable error less than 0.05%. 

III. ESTIMATION APPROACHES FOR STATE OF CHARGE

A. Coulomb Counting

The current is accumulated regarding the time in this algorithm.

\[
\text{SOC}(t) = \text{SOC}_0 + \frac{1}{Q} \int_0^t \text{Idt}
\]

(14)

The beginning state of charge is \( \text{SOC}_0 \), and the total state of charge is \( \text{SOC} \). \( I \) is the charging/discharging current and \( Q \) is nominal capacity of the battery.

B. EKF(Extended Kalman Filter) algorithm

The Li-ion batteries are completely nonlinear, and their characteristics are changed all the time. For example, dimensional OCV curve in line with State of Charge is completely nonlinear. Furthermore, due to changing operating conditions, some essential EECM (Equivalent electrical circuit model) metrics, such as polarization resistance and capacitance, change by time. Therefore, Linear Kalman Filter is useless to estimate the SOC of these kinds of batteries. It is required to linearize all parameters including OCV-SOC around their functioning spot [9]. Nevertheless, the OCV in most Li-ion batteries needs a long time to settle in the correct value and we should take it time before measuring and having precise SOC-OCV curve and in some cases, there is not enough time to put.[10] The Extended Kalman Filter (EKF) has two main transition and measurement equations that should be linearize. If the linear approximations are accurate enough, it can be employed for nonlinear circumstances. First order Taylor approach is applied for linearizing transition and measurement equation by Jacobians. Process noise and observation noise assume being Gaussian [11]. The transition equation’s distribution of propagating states is estimated as a Gaussian PDF. Measurement equations are estimated to be Gaussian as well.

The nonlinear system’s equations are as follow:

\[
x(n+1) = f(w(n), u(n), x(n)) \quad (15)
\]

\[
y(n) = h(v(n), x(n)) \quad (16)
\]

(a) Prediction stage:

\[
x_a(n+1) = f(u(n), x_e(n)) \quad (17)
\]

\[
M(n+1) = W(n) \sum_w W(n)^T + F(n)P(n)F(n)^T \quad (18)
\]

(b) Correction stage:

\[
K(n+1) = V(n) \sum_v V(n)^T \quad (19)
\]

\[
P(n+1) = -H(n)K(n+1)M(n+1) + M(n+1) \quad (20)
\]

\[
x_e(n+1) = x_a(n+1) - h(x_e(n+1), y(n+1)) \quad (21)
\]

To predict the future state, the transition equation is used straightforwardly. The related covariance matrix \( M \) is constructed by Jacobians with \( x_e(n), w(n) \) assessment:

\[
H(n) = \frac{dh(v, x)}{dx} \quad F(n) = \frac{df(w, x)}{dx} \quad (22)
\]
Covariance matrix P is updating by the following Jacobians that is evaluated at $x_a (n + 1), v(n)$:

$$V(n) = \frac{dh(x,v)}{dv} \quad W(n) = \frac{df(x,w)}{dw}$$

The estimation error is directly computed using the measurement. Comparing the prediction and update step equations in Standard Kalman Filter with these steps in EKF, we can say that the Jacobian F(n) is said to serve the role of the A matrix, while the C matrix is the Jacobian H(n).

P is the state covariance, W is the process noise covariance matrix, and V assumed as the measurement noise covariance matrix. Using data taken from sensor (y) and Kalman gain, the Kalman filter will decide to rely on estimation or measurement. As a result, Kalman filter will update the estimation while, y is the measurement taken from the sensor.

Observation noise and process noise directly affect errors and RMSE. Therefore, by tuning the values the best amount for them set as follows:

$$Sw = \begin{bmatrix} 1.0e^{-6} & 0 & 0 \\ 0 & 1.0e^{-5} & 0 \\ 0 & 0 & 1.0e^{-5} \end{bmatrix} \quad Sv=10^{-5}$$

Figure 3. a. SOC estimation by EKF and formal Coulomb Counting method b. V1 estimation by EKF c. V2 estimation by EKF

Figure 4. Measured and estimated battery terminal voltage

Figure 5. Estimation error of terminal voltage

Figure 6. Battery internal Resistances
IV. CONCLUSION

In this study, SOC and terminal voltage estimation done with the EKF algorithm Li-ion batteries. The state space analysis is done by applying related equations on the 2 RC branch battery model. The Hybrid Pulse Power Characterization (HPPC) test data obtained at several temperature from 40°C to -10°C are used to calculate the SOC 3-dimensional curve as a function of SOC and T. The simulation results show that the EKF is more productive and precise than C_C (Coulomb Counting). As a result, this method is more reliable method. The inaccuracy in the EKF results is less than 1%, indicating that EKF is a trustworthy method for estimating battery states. Moreover, RMSE is utilized for assessment index to quantify the estimation error of techniques. RMSE values for Terminal Voltage and SOC are 5% and 1.7% respectively.

REFERENCES

[1] Kollmeyer, Phillip; Skells, Michael (2020), “Turnigy Graphene 5000mAh 65C Li-ion Battery Data”, Mendeley Data, V1, doi: 10.17632/4fs8cjpq5m.1
[12]
Dynamic Edge-Based Graph Attention Network and Visual-Semantic Embedding for Outfits Recommendation

Rucong Xu, Jianfeng Wang, Yun Li*
Shenzhen Institute for Advanced Study
University of Electronic Science and Technology of China
Shenzhen 518110, China
R.C.Xu@i4ai.org, J.F.Wang@i4ai.org, Yun.Li@ieee.org

Abstract—Currently, outfit compatibility matching used in fashion e-commerce has been based on visual and textual information to learn global features from items to choose from, which has ignored local feature relationships between spatially adjacent items. We propose a new method that combines a dynamic graph neural network and visual semantic embedding to effectively discover the latent fashion compatibility. It takes visual and semantic information of items as input and extracts global and local information of items through the network to learn the compatibility of the outfit. This method enhances the local connections of items in an outfit by capturing the relationship between items, thus revealing potential information about the outfit collocation at multiple dimensions. Experimental results conducted on a real-world Polyvore dataset show that our proposed approach outperforms all the state-of-the-art methods in both accuracy and efficiency.

Keywords—Visual-Semantic Consistency; Graph Attention Network; Outfit Recommendation

I. INTRODUCTION

With the boom of the online economy, there has been a growing demand for online shopping of fashionable clothing and outfits. As online shopping only provides a picture of the fashionist’s virtual try-on clothing or even the original picture of the outfit, and not all users have a keen sense of fashion, selecting the right outfit to match has become a major challenge in online clothing shopping. An outfit is often a combination of several fashion items, and the compatibility of the items and overall style coordination determines whether an outfit is compatible or not. As a result, it is a major part of fashion recommendations to measure the compatibility of an outfit and recommend more suitable fashion items to the user.

There has been research progress in clothing recommendation and outfit matching, but most of the current approaches either only consider the compatibility of the image and description information of a set, or treat the items as nodes and the set as a network of interconnected nodes to learn the compatibility between items [4][5][7]. However, the style of an outfit is determined not only by its attributes such as colour, material and shape of the item but also by the style compatibility between items, especially between close neighbouring items [1]. In particular, there is lack of a visual feature extraction module and a dynamic graph attention module. These are necessary to learn the attributes of the items in an outfit and the latent relationships between them, thus leading to high compatibility of items.

To diagnose the compatibility of an outfit and improve the overall compatibility of an outfit by recommending more compatible items, we propose a model that combines a dynamic graph attention network and visual-semantic embedding. In this work, we aim to

- develop a model that considers a graph neural network and visual-semantic embedding to extract local and global information from images of fashion items, as well as descriptive textual information about the items, in such a way as to learn the potential information for multi-level clothing matching;
- develop a dynamic edge-based graph attention module to learn style matching relationships between items in close style similarity, allowing graph neural network to better learn the compatibility of relationships between items in latent spatial proximity; and validate the model on the Polyvore outfits dataset for visual and semantic information of the outfit in recommending the most compatible items.

II. RELATED WORK

Several important theoretical approaches and models have been proposed by researchers to address the compatibility of outfits. These models have been explored in several ways, starting with a spatial metric-based learning approach, which focuses on measuring the compatibility between items based on distance in space, mapping items to style space, and calculating the distance between feature vectors of items in style space. Considering the ambiguity and breadth of the definition of compatibility, Yong et al. mapped each item to $K$ subspaces in the latent space and performed the distance between the metric item quotient vectors in the subspaces, ensuring that there is at least one subspace in the potential space where the distance between items in the space is compatibility is determined [8]. In this way, the diversity of
compatible items is ensured. The second type of approach is based on a combination of sequence models to generate outfit pairings, which sorts the items within an outfit into sequences and uses LSTM sequence prediction for the next item. Vasileva et al. model the compatibility between items by jointly optimizing the embedding of items and category-specific complementary relationships in a unified space [7]. Han et al proposed the BILSTM network, which defines an outfit as a sequence, models the items within the outfit as sequential relationships, and uses a bidirectional LSTM network to predict the next item after a given set of items by employing a recurrent neural network to model the style of the outfit as a continuous process, and the compatibility score of the outfit [5].

To better capture the compatibility of items within an outfit, researchers proposed a combination generation method based on graph neural networks to match items. Cui et al. directly used category information to map items into the corresponding category nodes in the fashion graph, where items of the same category are often close to each other in the visual space. predict the relationship between items [9]. These approaches learn compatibility through multi-dimensional features, which can increase the accuracy of the model. However, in the user shopping process, fashion items contain rich information and compatibility between items needs to consider not only the overall style compatibility but also the style compatibility of neighbouring items. Compatibility between clothing items is not only determined by the attributes of these items themselves but is also influenced by their tastes and the fashion sense of the items, requiring consideration of the relationship between the user’s interests and outfit matching. In contrast to previous research approaches, this paper takes a holistic and local perspective, using a combination of a dynamic edge-building graph attention network and visual semantic embedding to model clothing compatibility through visual information about items and category interactions between items, as well as the compatibility of visual semantic information.

III. MODEL DESIGN

Our approach aims to perform compatibility computation for equipment of different appearances, the overall model design is illustrated in Fig. 1. The features extracted from each item are compared in the feature comparison module to calculate the relationship matrix between the items, and the outfit compatibility prediction module is used to calculate the compatibility score from the relationship matrix. The visual semantic embedding module is used to calculate the consistency of the image features with the text features.

A. Feature Extraction Module

In this part, we define a set of fashion items as $s = \{s_1, s_2, s_3, \ldots, s_i\}$ and each outfit is associated with a set of $i$ fashion items $s$, denoted as $o = \{o_1, o_2, o_3, \ldots, o_n\}$. In addition, each item contains relevant text description which can be indicated as $X = \{x_1, x_2, x_3, \ldots, x_i\}$, where $x_i$ is $i$-th text description of $i$-th fashion item $s$. The visual embedding can be obtained as follows:

$$f_i^L, f_i^2, \ldots, f_i^L = \text{CNN}(s_i; \theta),$$

(1)

where $f_i^L \in R^{d}$ is visual feature of item $i$ at the $l$-th convolutional layer, $d$ is its dimension and $\theta$ refers to the parameters of the CNN network. The pre-trained resnet18 is used as our CNN network.

For the visual features in the 1 to $l$-th layers, these are sent to the Global Attention Mechanism (GAM) which can assign weights to pixels based on the importance of the pixel points in the feature, to enhance the embedding representing of the feature. Using the GAM module allows the model to focus more on the area of the picture where the item is located. In particular, the Global Averaging Pooling operation (GAP) is used to summarize the visual representations, this operation shows significant performance in the extraction of attributes from items. The enhanced visual features can be obtained as follows:

$$F_i^L = \text{GAP}\left(\text{GAM}\left(f_i^L\right)\right),$$

(2)

where $F_i^L \in R^{d_m}$ is the final visual feature of item $i$ at the $l$-th convolutional layer and $d_m$ is its dimension. Fig. 2 shows the visualization of attention maps (heatmaps) for the fashion items. It can be seen that the GAM module allocates a higher weight to the area of the image where the item is located.

Each outfit is constructed from various items, and the compatibility of an outfit depends on the relationship of the items to each other. In order to extract the relationships of the embedding of the item, we establish a graph network $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where $f_i^L \in \mathcal{V}$, $f_i^L$ is the visual embedding of $s_i$ at the last convolutional layer. $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$ is the set of edges between each $\mathcal{V}$. Graph attention network (GAT) imposes attention on the network by giving higher weight to more relevant items of an outfit in the neighbourhood aggregation process. The graph-based visual features can be obtained as follows:

$$F_i^L = [f_i^L, \text{GAT}\left(f_i^L, \mathcal{E}\right)],$$

(3)

where $F_i^L \in R^{d_f}$ is the final visual feature of item at last convolutional layer, $d_f$ is its dimension and $[.]$ denotes the concatenation operation. To validate the effectiveness of our feature extraction model, we visualise some final extracted feature vectors $F^L$ with t-SNE [11] in Fig. 3. We observe that similar items in the embedding space have similar features such as type, colour, style, and texture, which is a good demonstration of the effectiveness of our image feature extraction module.

B. Dynamic Edge-Based Graph Network

The relationships between items within the same outfit can be represented using a graph network. Different from the traditional edge building for all nodes, we resort to the feature embedding similarity, due to semantic embeddings are potential representations of shape, style, and colour, and items with higher semantic embedding similarity contribute
Fig. 1. Architecture of the proposed model, which combines a dynamic graph neural network and visual semantic embedding. The feature extraction module is used to extract features from the items.

Fig. 2. Visualization of attention maps (heatmaps) for the fashion items.

more to the overall style than items with lower similarity. Accordingly, edges are created using the feature embedding similarity between features extracted from different outfits, we can attain the edge $E$:

$$E^{i,j} = J\left(d\left(f^L_i, f^L_j\right)\right),$$

(4)

where $E^{i,j} \in V \times V$ is the edge between item $i$ and item $j$, $E^{i,j} = 1$ means there is an edge between item $i$ and item $j$, $E^{i,j} = 0$ means no edge between item $i$ and item $j$. $d\left(\cdot\right)$ is the cosine similarity function, it can calculate the feature embedding similarity between $f^L_i$ and $f^L_j$. $J\left(\cdot\right)$ will judge the value of the similarity and make $E^{i,j} = 1$ when the similarity is above $p$, and 0 otherwise.

C. Visual Semantic Embedding Module

A set of outfits is usually represented by multimodal information such as images, and text. Visual semantic embedding (VSE) is a way of providing a common representation between different modalities so that VSE can be used to make the most of information from different modalities.

For the $i$-th item in one outfit $o$, its text description $x_i$ can be represented as one-hot vector $e_i$. The word embedding of $e_i$ is $w_i = W_T e_i$ and the semantic embedding of an outfit is $v = \frac{1}{M} \sum_{i=1}^{M} w_i$, where $M$ is the number of items in the outfit and $W_T$ is the weight of word embedding model. Similarly, the visual feature $f^L$ is mapped into the embedding space as $u = W_F f^L$, where $W_F$ is the weight of visual embedding model. The goal of VSE is to make $u$ and $v$ close to each other in the joint space when they come from the same item.

The loss function is obtained as the follows:

$$L_{vse}(v, u) = \sum_u \sum_k \max(0, m - d(u, v) + d(u, u_k)) + \sum_v \sum_k \max(0, m - d(v, u) + d(v, v_k)),$$

(5)

where function $d\left(\cdot\right)$ is used to calculates the distance between embedding $u$ and embedding $v$. $u_k$ is the visual embedding that is not compatible with item $u$ and $v_k$ is the semantic embedding that is not compatible with item $v$.This function expects the distance between all matched pairs $u$ and $v$ to be smaller than all of the other not compatible items with a margin $m$. 

213
D. Feature Comparison Module

Different items in the same outfit have corresponding feature values in different convolution layers. For each layer, we compute the features obtained for each item in relation to the other items once. We let $r_{ij} = f(f^l_i, f^l_j)$ to denote the similarity between item $i$ and item $j$.

$$f(f^l_i, f^l_j) = d(max((f^l_i \otimes m(i, j), f^l_j \otimes m(i, j), 0)))$$  \hspace{1cm} (6)

where vector $m(i, j)$ is with the same dimension as the features $f^l_i$, $\otimes$ indicates the element-wise product operation. The mask $m(i, j)$ functions as an element-wise gating function which select the relevant elements of the feature vector for different compatibility conditions. For the $l$-th layers, the similarity between features can be expressed in matrix form,

$$R^l = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N} \\ r_{21} & r_{22} & \cdots & r_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N1} & r_{N2} & \cdots & r_{NN} \end{bmatrix},$$  \hspace{1cm} (7)

where $R^l$ represents the similarity matrix between items at layer $l$ and $r_{ij}$ represents the similarity between item $i$ and item $j$.

We stitch the matrix obtained at each layer to obtain the final relationship vector of the outfit:

$$E = [R^1; R^2; \ldots; R^l],$$  \hspace{1cm} (8)

where $[; ; ; ]$ denotes concatenating multiple matrices and flattening them into a vector, $E \in R^{dh}$ is the final relationship vector of the outfit and $dh$ is its dimension.

E. Outfit Compatibility Prediction Module

For each set of outfit relations, they receive multiple hidden complementary factors, such as texture, colour, style, etc. We wish to map them to multiple complementary factor subspaces to obtain a richer feature representation. We use a multi-head MLP predictor to initialise the relational representations of different features with different latent subspaces which are defined as,

$$E^k_j = W^k_b E_j,$$  \hspace{1cm} (9)

where $E_j \in R^{dh}$ is the initial relationship embedding of outfit $j$, $W^k_b \in R^{dh \times dh}$ is the weight matrix for transforming the relationship embedding of each outfit into the $k$-th hidden factor space, $E^k_j \in R^{dh}$ denotes the $k$-th hidden factor of outfit representation.

It should be noted that in the absence of constraints, the hidden factors of the relationship embeddings tend to converge, leading to unnecessary compatibility inference across different branches. To enable different branches to learn the hidden factors of different relationship embeddings, we introduce a complementarity regularisation based on orthogonality. Formally, we obtain the objective function,

$$L_m(s) = \sum_{j=1}^{m} \| E_j E_j^T - I \|_F^2,$$  \hspace{1cm} (10)

where $I \in R^{K \times K}$ is the identity matrix. $E_j = [E^1_j; E^2_j; \ldots; E^k_j] \in R^{dh \times dK}$ denotes different relationship embedding of outfit $j$ and $\| \cdot \|_F^2$ denotes L2 regularization.
Ultimately, for each hidden factor embedded in different relationships, the MLP with two layers was empirically selected to calculate the compatibility score corresponding to each hidden factor for item $s$. The scores of all hidden factors were eventually averaged to attain the final compatibility score $\hat{y}$ for outfit $O$ as follows,

$$\hat{y} = \frac{1}{K} \sum_{k=1}^{K} \sigma (W_2 (a (W_1 E^k + b_1)) + b_2),$$  \hspace{1cm} (11)

where $W_1 \in R^{d_k \times d_0}$, $W_2 \in R^{d_0 \times 1}$, $b_1 \in R^{d_0}$, $b_2 \in R^1$ are the parameters of the MLP, where $d_k$ is the dimension of $E^k$, and $d_0$ is the number of hidden units of the MLP. $\sigma$ is the sigmoid active function, it regards the compatibility score as the probability that the outfit is compatible.

We resort to the binary cross-entropy loss to optimize the model, which shows the great superiority in the classification tasks,

$$L_{clf} = -y \cdot \log(\hat{y}) - (1 - y) \cdot \log(1 - \hat{y}),$$  \hspace{1cm} (12)

where $\hat{y}$ is the estimated score and $y$ is the ground truth label.

Inspired by [4], the model is optimized by using two additional losses to penalize the training: $L_{mask}$ aims to make the masks to be sparse and $L_{emb}$ enables the CNN to encode normalized features in the latent space.

$$L_{mask}(m) = \| m \|_1,$$  \hspace{1cm} (13)

$$L_{emb} (F^L) = \| F^L \|_2.$$  \hspace{1cm} (14)

Therefore, the final loss function can be developed as follows:

$$L_{total} = L_{clf} + \lambda_1 L_{vse} + \lambda_2 L_{emb} + \lambda_3 L_{mask} + \lambda_4 L_m,$$  \hspace{1cm} (15)

where $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are the weights for each additional losses.

## IV. EXPERIMENTAL TESTS

### A. Experiment Settings

For the outfits in Polyvore, we build a model that combines a dynamic graph attention network and visual-semantic embedding for outfit compatibility prediction. The pre-trained resnet18 is used to extract the image features of the items. All experiments were trained on an RTX 3090 GPU with a batch size of 32, taking a total of 22 hours to train 100 epochs. We set the weights of additional loss $\lambda_1, \lambda_2, \lambda_3, \lambda_4$ are 1, 5e-3, 5e-4, 5e-3, $p = 0.5$, $K = 4$ and $m = 0.2$. The optimizing strategy is SGD with a momentum of 0.9. We save the model parameters that perform best on the validation set.

### B. Model Comparison

Our model is trained on the Polyvore dataset. We filtered categories that were not relevant to clothing to end up with 158 categories, after which we divided these 158 categories into 5 groups (top, bottom, shoe, bag, accessory), resulting in a dataset of 16,176 training samples, 2,463 test samples, and 1,196 validation samples.

For our model, we set two evaluation metrics outfit compatibility prediction (AUC) and fill-in-the-blank (FITB). The goal of the Outfit Compatibility Prediction task is to compute a score as overall compatibility, the Fill-in-the-blank task is to select the item that best matches the outfit with missing items, which is evaluated by the accuracy of the correct answers to the questions. Therefore, we evaluate the performance of the model using FITB Accuracy and CompAUC. To evaluate our method, a detailed comparison with existing methods was made. The evaluation results on the two metrics are shown in Table I. It can be shown that our method is the most advanced model for both the AUC and FITB evaluation metrics.

### C. Ablation Experiment

To analyse the validity of the modules in the model, we perform an ablation experiment. The results of the experiments are shown in Table II. From the table, we can see that the other graph neural networks are not as effective as the graph attention network and the fully connected edge building is not as effective as the dynamic edge building. In particular, the Model with Visual Semantic Embedding(VSE) module, Global Attention Mechanism(GAM) obtains a better representation of compatibility.

We then explore the effect of the value of $K$ in the multi-head MLP predictor on the model in both the compatibility prediction task and the fill-in-the-blank task. From the Table III, the performance of the model does not increase monotonically with the number of predictors, but first increases to a maximum value at $K = 4$ and then decreases as the number of $K$ increases further. This demonstrates that the best
performance of our multi-headed MLP predictor is achieved only for $K = 4$. Adding too many predictors may add noise and lead to performance degradation.

D. Case Study

For a given outfit, we select the least compatible items in the outfit and choose the most appropriate items to replace them to improve the overall compatibility of the outfit. The diagram shows the process of replacing the incompatible items in our outfit. It can be shown in Fig. 4 that the compatibility score for the first outfit changed from 0.6802 to 0.9566 and for the second outfit from 0.1055 to 0.9335, with the earthy cotton jumper in the first outfit being more in keeping with the overall style than the original bright leather coat, and the light canvas bag in the second outfit being more fashionable than the original dark leather bag.

V. Conclusion

In this paper, we develop an outfit compatibility matching model that makes use of the combination of a dynamic graph attention network and visual-semantic embedding. It dynamically extracts visual and semantic information about the outfit and the relationship between items in the outfit. The dynamic edge-based graph attention module enhances the local connections of items in an outfit by capturing deep connections between items and learning potential information about the outfit collocation at multiple dimensions. Experimental results show that this approach is more accurate and effective than existing approaches.

In the future, we will evaluate the model on more larger datasets by increasing the number of different kinds of samples to make the model more generalisable. In addition, we will check the model on a large e-commerce platform to further test the online performance of the model.

Acknowledgment

This work is supported in part by the Guangdong Natural Science Foundation under Grant 2021B1515120078.

References


Image segmentation of micro-TIG battery welds

Carlo Ferri∗, Malarvizhi Kaniappan Chinnathai†, Rohin Titmarsh ‡ and Hala Abdelaziz§

∗‡§ WMG, The University of Warwick
6 Lord Bhattacharyya Way, Coventry, CV4 7AL, UK
∗ ORCID: 0000-0002-2362-0312
Email: ∗ c.ferri@warwick.ac.uk, † rohin.titmarsh@warwick.ac.uk, § hala.abdelaziz@warwick.ac.uk
‡ Brunel Innovation Centre, Brunel University
TWI, Granta park, Great Abington, Cambridge, CB21 6AL, UK
Email: m.kaniappan-chinnathai.2@warwick.ac.uk

Abstract—Inspection of cell-to-tab welds in module assembly for battery pack is one the most critical processes in the production of battery packs for transportation electrification. A procedure suitable for the segmentation of weld images on module assembly lines is proposed to separate them into weld and tab regions. The procedure is centred around identifying the edge of the weld and the convex hull region that includes it. The edge is detected with a fuzzy logic rule-based inference system. The procedure is demonstrated with a set of 71 images that are labelled to establish a comparison reference, the so called ground truth. Particle swarm optimisation is used to find values of the parameter procedure that result in a a local minimum of the mean percent error (MPE). An MPE of 4.25 per cent has been obtained.

Index Terms—transportation electrification, fuzzy logic, particle swarm optimisation

I. INTRODUCTION

A number of segmentation approaches have been proposed in the detection of weld defects in radiographic images. Among these approaches, Malarvel et al. [1] proposed an improved Otsu’s method that automatically selects the threshold value using the Weibull distribution. Rathod and Anand [2] compared three segmentation methods: the edge- and boundary-based method, the region-growing method and the watershed transformation method. Mahmoudi and Regragui [3] developed a segmentation approach based on contrast enhancement, histogram analysis, global image thresholding using the Otsu’s method and local image thresholding using the Sauvola method. Carrasco and Mery [4] presented a weld defect detection method that comprises edge detection, binary thresholding and the application of watershed transform.

In welding robotic systems, vision sensors have been used to track the weld seam. Haldankar et al. [5] have reviewed software architectures and methods for weld images in real-time seam tracking, including segmentation. For gas metal arc welding (GMAW), Xu et al. [6] proposed a method for identifying the weld seam in real-time using the Otsu’s method and edge detection filters.

Sun and Li [7], in surveying meta-learning algorithms for image segmentation, have highlighted the challenge of applying segmentation methods based on deep learning in those applications where a large amount of labelled data is unavailable. Sultana et al. [8] have surveyed the evolution and the state-of-the-art of image segmentation methods using deep convolutional neural networks. In prototype and small-series production of battery packs for transportation electrification and special applications, Micro-TIG welding is often used for joining cells to tabs. In this production scenario, labelled images in large amount are often hardly available.

Moreover, the image characteristics of the joints may vary significantly from a small series to another. For example, a batch may have copper tabs, which are red-orange in colour, another nickel-coated tabs, which are grey. Therefore a segmentation method should adapt easily to variations between batches to be of greater practical use.

Segmentation methods of weld images specifically designed for the automatic micro-TIG welding of cells to tabs in small-series production have not been found in the literature. In the detection of blood vessels in retinal fundus images, Orujov et al. [9] demonstrated the flexibility of fuzzy logic edge detection by evaluating the universal method that they have proposed on three different image datasets. They have also showed that the accuracy of their method is significantly better than the performance of the classical methods for boundary detection specifically developed for each of the three datasets. Classical methods that they considered for comparison were the Laplacian, the Robert, the Robinson, the Scharrm, and the Sobel methods.

An approach to micro-TIG image segmentation based on fuzzy logic edge detection and particle swarm optimisation (PSO) is proposed in this paper.

In Section II the proposed segmentation method is introduced. In Section III, the particle swarm optimisation variables and the fuzzy edge detection parts of the segmentation method are defined. In Section IV, numerical results are presented. Conclusions are then drawn in Section V.

II. SEGMENTATION

The proposed segmentation procedure consists of the steps in the subsections that follows and that are described using the symbols and the conventions for system flowcharts prescribed
A. Taking weld colour images

A set of colour images of micro-TIG welds made in stable process conditions is formed. The number of images is sufficiently large to make the set representative of the whole future weld production in the same stable conditions.

B. Image downscaling

The images are scaled down to a number of pixels that makes the processing time compatible with the short cycle time of battery module assembly. The aspect ratio of the original images is preserved. The parameters of this image resizing are not part of any optimisation.

C. Pixel labelling

The pixels of the downscaled images are labelled as weld and tab. They are marked as one and zero, respectively. The tab pixels constitute the background, the weld pixels constitute the foreground or region of interest. A mask set, i.e. a set binary images, is the output of the labelling process and is designated with GT. The pixels labelled as weld are the ground truth. The meaning of the expression ground truth used here is ‘information obtained by direct visual examination, esp. as used to check or calibrate an automated recognition system’ [11]. Pixel labelling is therefore performed by a person using interactive software tools. Examples of these tools have been surveyed by Sagar et al. [12].

D. Converting images to grayscale

Each colour image is separated into its three colour components in the RGB (red, green, blue) and in the HSV (hue, saturation, value) colour spaces. Each colour image is also converted to grayscale by eliminating the hue and the saturation while retaining the luminance. The grayscale pixel value is obtained by the following weighted sum of the red, green and blue components:

\[ E'_y = 0.2989E'_R + 0.5870E'_G + 0.1140E'_B \]  

(1)

The weights in (1) rounded to three decimals are the same as those used to calculate the luminance \( E'_y \) in the recommendation ITU-R BT.601-7 [13], page 4). In this way, seven grayscale image sets are obtained and designated as \( G_i \) with \( i = 1, \ldots, N \) and \( N = 7 \). The procedure is run separately on each of the seven grayscale image sets obtained.

E. Particle swarm core fuzzy procedure

A software procedure based on the detection of edges and the computation of their convex hulls returns a set of masks designated as \( C_i \). Each set \( C_i \) corresponds to the input grayscale images in the set \( G_i \). As in the ground truth mask set, weld pixels are marked with one and tab pixels with zero.

F. Computing the MPE

The per cent error between the computed mask and the ground truth is calculated for each mask in the output from the core procedure, \( C_i \), and the corresponding mask in the ground truth set, GT. The mean per-cent error over the computed mask set \( C_i \) is then calculated and designated as \( \text{MPE}_i \).

As a part of the core procedure, a population of values for the vector of optimisation variables is generated randomly in the domain of the function under optimisation. The population is referred to as a swarm. Each vector in the swarm is called a particle, although the creators of the algorithm also discuss the term point, before excluding it [14]. For each particle in the swarm the mean per-cent error, i.e. the MPE, is then calculated. The particle yielding the minimum MPE is selected as the candidate local optimum. Together with the swarm, a vector of particle change is also calculated for each particle. The vector of changes is called velocity vector. The availability of a velocity vector allows an iterative process to take place: the velocity vector is added to the initial swarm of particles so that a second swarm of particles is obtained. As the names suggest, the second particle swarm can also be seen as a displacement of the first swarm during a dimensionless time step, i.e. during an iteration. The generation of new swarms continues until a stopping criterion is met. In the proposed procedure the stopping criterion consists in obtaining a proposed local minimum MPE that has not changed more than \( \text{P} \% \) in the last \( \text{T} \) iterations.

G. Selecting the locally-optimal mask set

The set of computed masks \( C_i \) that yields the minimum per cent error over the seven grayscale image sets \( G_i \) is selected.

H. Overlaying the masks to the input colour images

Each mask in the locally optimal mask set found is overlaid to each of the RGB or HSV components of the colour images in the downscaled colour image set. The weld colour segments are therefore obtained. Overlaying occurs by element-wise multiplication, a.k.a. Shur product.

III. PARTICLE SWARM CORE FUZZY PROCEDURE

The particle swarm core procedure consists of a sequence of processes described in the subsections that follow and illustrated with a flowchart in Fig. 3.

A. Setting values to optimisation variables

The variables whose values is tuned to determine a local minimum for the MPE are as it follows:

- \( X_1 \): The standard deviation of the Gaussian kernel used to filter each grayscale image set.
- \( X_2 \): The standard deviation of the Gaussian membership function defining the fuzzy set \( \text{zero}_H \) for the horizontal gradient in the grayscale images.
- \( X_3 \): The standard deviation of the Gaussian membership function defining the fuzzy set \( \text{zero}_V \) for the vertical gradient in the grayscale images.
Fig. 1. System flowchart of the proposed procedure.

Fig. 2. (a) A weld colour image. (b) A ground truth mask. (c) Grayscale image in the blue set. (d) Computed locally-optimal mask. (e) Locally-optimal weld segment.
Fig. 3. Flowchart of the particle swarm core fuzzy procedure.

$X_4$: The value identifying the beginning of the triangular membership function defining the fuzzy set $\text{edge}$ having the other two vertices in $(1,0)$ and $(1,1)$.

The standard deviation $X_1$ is further described in Section III-B. The remaining three variables $X_2, X_3, X_4$ are a part of the edge detection procedure based on fuzzy logic and described in Section III-C.

B. Image filtering

Each image in the set $G_i$ is filtered with a Gaussian filter, whose standard deviation in pixels, $X_1$, is the first variable considered in the minimisation of the $MPE$. A square filter kernel is used, whose side length in pixels, $f$, is set as a function of the standard deviation as displayed in (2), where the symbol $\lceil X \rceil$ represents the smallest positive integer greater than $x$.

$$f = 2\lceil 2X_1 \rceil + 1 \quad (2)$$

When convolving an image with the filter kernel, any needed padding at the image border is performed by repeating the border pixels.

C. Fuzzy logic edge detection

The edges of the weld segment in each filtered image in input to this step are detected by a rule-based fuzzy inference system (FIS). First, the gradients of the pixel luminance in the horizontal and in the vertical directions are calculated. The following two rules, $R_1$ and $R_2$, are then used to compute the degree of membership of each input image pixel:

$R_1$: If $\text{horizontal\_gradient}$ is $\text{zero\_H}$ and $\text{vertical\_gradient}$ is $\text{zero\_V}$ then $\text{FIS\_output\_pixel}$ is not $\text{edge}$.

$R_2$: If $\text{horizontal\_gradient}$ is not $\text{zero\_H}$ or $\text{vertical\_gradient}$ is not $\text{zero\_V}$ then $\text{FIS\_output\_pixel}$ is $\text{edge}$.

The labels $\text{horizontal\_gradient}$ and $\text{vertical\_gradient}$ represent the two universes of discourse on which the two fuzzy sets $\text{zero\_H}$ and $\text{zero\_V}$ are defined. The meaning of the expression $\text{universe of discourse}$ is taken from the Oxford English Dictionary as ‘all those objects, elements, etc., that the terms of a proposition may refer to; a universal set; the totality of things under consideration’ [15]. The fuzzy sets $\text{zero\_H}$ and $\text{zero\_V}$ have a suffix in their designation just to distinguish the universe of discourse to which each of them refers to. The label $\text{zero\_H}$ indicates the class of horizontal gradient values close enough to zero to be considered indistinguishable from it. The label $\text{zero\_V}$ indicates the same for the vertical gradient. The vagueness in the definition of these two classes is conveniently described with ‘a continuum of grades of membership’ [16]. Gaussian membership functions centred around zero and with standard deviation $X_2$ and $X_3$ are used to represent the grade of membership to $\text{zero\_H}$ and $\text{zero\_V}$, respectively. The expression of these two functions is in (3).

$$\mu_{\text{zero}_j}(x) = \exp \left( -\frac{x^2}{2X_k^2} \right) \quad j = H, V; \quad k = 2, 3 \quad (3)$$
where the symbol $\mu_{\text{zero}_j}(x)$ indicates the grade of membership of the gradient value $x$ to the set zero$_j$ with $j = H, V$. The subscript $k$ equals two if $j = H$ and equals three if $j = V$. The label $\text{FIS\_output\_pixel}$ represents the universe of discourse on which the fuzzy set edge is defined. This universe is constituted by the real numbers in the interval $[0, 1]$. The triangular membership function in (4) and (5) characterised by the parameter $X_4$ is used to model the edge fuzzy set.

$$\mu_{\text{edge}}(x) = 0 \text{ if } x \leq X_4;$$  \hspace{1cm} (4)

$$\mu_{\text{edge}}(x) = \frac{x - X_4}{1 - X_4} \text{ if } x > X_4;$$  \hspace{1cm} (5)

The triangular membership function is displayed in Fig. 4.

The evaluation of the two rules in the FIS allows the computation of the degree of membership to the fuzzy set edge. Every pixel in each image in output to the FIS is initially assigned a zero value. As a result of the FIS, this computed degree of membership is given as a value to the pixels in the output images. Therefore a pixel value one, i.e. a white pixel, describes an image part that is not an edge. A black pixel, describes an edge known with high confidence, a white pixel, describes an edge known with high confidence. All the grayscale values in between express the level of confidence that the pixel in question may be an edge.

The grayscale image set in output from the edge detection process is transformed to a binary zero-one image set by the Otsu’s method. This method selects the threshold value $T_i$ that minimizes the intraclass variance of the thresholded black and white pixels. By thresholding, the uncertainty about the attribution of pixels to the edge class or the not-edge class is eliminated.

E. Convex hull computing

For each set of white pixels describing an edge a convex hull is calculated. Each convex hull is marked with white pixels. This process makes it possible to identify those areas of the images that are inside an edge as weld regions.

F. Blob removal

Some of the pixel groups identified by computing convex hulls are too small to have any practical importance in the welding process. Those convex hulls made of less than an empirically-established number of pixels are set to zero. For the image set considered, the lower limit for the pixel number was set to 150 pixels.

IV. RESULTS

The Particle swarm core fuzzy procedure and the $MPE_i$ calculation in Fig. 1 have been implemented in a program using MATLAB and the MATLAB Parallel Computing Toolbox. The program has been run separately for each grayscale image set $G_i$ on a Unix-type high performance computer system (HPC) with 800 cores, of which 91 were used in a parallel cluster to execute the program.

In the stopping criterion described in Section II-F, the parameters $P\%$ and $T$ have been set to $1\%$ and 20 iterations, respectively. This choice has made it possible for the program execution on each of the seven $G_i$ image datasets to terminate always within the maximum number of iterations set to 800, when the number of particles in each iteration is set to 200. That is, with the swarm size set to 200, the program executions have always terminated within 800 swarm generations.

In Tab. I the local minimum $MPE_i$ values for each of the considered image sets $G_i$ is displayed. The blue image set yields the smallest $MPE_i$ value. The corresponding computed mask set is selected. The actual number of iterations with a swarm size of 200, the number of $MPE_i$ evaluations and the program run time on the above-described system are also displayed.

<table>
<thead>
<tr>
<th>$G_i$</th>
<th>$MPE_i$</th>
<th>Iteration count</th>
<th>Evaluation count</th>
<th>Run time, second /s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red</td>
<td>4.96</td>
<td>33</td>
<td>6800</td>
<td>2442</td>
</tr>
<tr>
<td>Green</td>
<td>4.42</td>
<td>35</td>
<td>7200</td>
<td>2659</td>
</tr>
<tr>
<td>Blue</td>
<td>4.25</td>
<td>36</td>
<td>7400</td>
<td>2346</td>
</tr>
<tr>
<td>Hue</td>
<td>9.66</td>
<td>40</td>
<td>8200</td>
<td>2685</td>
</tr>
<tr>
<td>Saturation</td>
<td>30.0</td>
<td>23</td>
<td>4800</td>
<td>1899</td>
</tr>
<tr>
<td>Value</td>
<td>4.95</td>
<td>38</td>
<td>7800</td>
<td>2550</td>
</tr>
<tr>
<td>Grayscale</td>
<td>4.69</td>
<td>35</td>
<td>7200</td>
<td>2553</td>
</tr>
</tbody>
</table>

The actual number of iterations in Tab. I is far below the maximum number of iterations that has been set: increasing this maximum does not change the $MPE_i$. The values of the parameters $P\%$ and $T$ have been set specifically to obtain this effect. Further $MPE_i$ reductions might be obtained by increasing the swarm size at the cost of an increased run time. Quantifying these potential reductions is beyond the purpose of this investigation.
V. CONCLUSIONS

A procedure for segmenting colour images of micro-TIG welds is proposed. The procedure aims to be part of weld inspection processes in battery module assembly lines. Fuzzy logic edge detection complemented by the determination of their convex hull appears a viable strategy for the stated aim. A demonstration of the implemented software programs on a set of 71 micro-TIG labelled images resulted in a local minimum per cent error of 4.25.

ACKNOWLEDGMENT

This investigation was performed as part of the research project NON-SLIP, Machine Learning for On-line Cell-to-busbar Weld Visual Inspection, funded by the UK program High Value Manufacturing Catapult at WMG, The University of Warwick.

REFERENCES


Real-Time Instance Segmentation of Pedestrians using Transfer Learning*

Bare Luka Žagar, Tobias Preintner, Alois C. Knoll
Chair of Robotics, Artificial Intelligence and Real-Time Systems
Technical University of Munich
Munich, Germany
{bare.luka.zagar@tum.de, tobias.preintner@tum.de, knoll@in.tum.de}

Ekim Yurtsever
Center for Automotive Research
The Ohio State University,
Columbus, OH 43212, USA
yurtsever.2@osu.edu

Fig. 1: Comparison of a synthetic and a real-world image. Left: a real-world image of an urban street scene from the Cityscapes [14] dataset. Right: a synthetic image of an urban street scene from the Synscapes [15] dataset.

Abstract—Real-time instance segmentation of pedestrians presents a critical core task within an automated driving pipeline. Recent research focuses on existing real-world datasets to train their instance segmentation networks. However, due to the limited size of real-world datasets, they tend to either overfit or lack accuracy. Therefore, these networks remain useless for real-world applications. Hence, we introduce a transfer learning strategy by combining a large-scale synthetic dataset and a real-world dataset for instance segmentation of pedestrians. We showcase our approach on three state-of-the-art real-time instance segmentation methods: (1) YOLACT++, (2) SipMask, and (3) BlendMask. Finally, we provide a quantitative and qualitative evaluation of our introduced approach on two publicly available urban street scenes datasets, i.e. the real-world Cityscapes dataset and the synthetic Synscapes dataset.

Index Terms—Real-Time Instance Segmentation, Transfer Learning, Automated Driving Systems, Synthetic Data

I. INTRODUCTION

Automated driving [1] could bring improved roadway safety and thus fewer vehicle accidents, according to a report [2] by the National Science Technology Council and the U.S. Department of Transportation. Moreover, the development of automated vehicle technologies (AVT) becomes more crucial if we consider that more than 94% of road accidents are due to human error [3]. Therefore, one essential safety aspect of AVT is the capability of accurate and real-time segmentation of pedestrians.

Instance segmentation [4], [5] is a core problem in the computer vision research field, and it can be seen as a combination of object detection and semantic segmentation [5]. Hence, instance segmentation mitigates the limitations of object detection and semantic segmentation, as it will not suffer from overlapping bound boxes or merged segmentation masks. Most of the recent research in instance segmentation [6]–[10] does not consider the aspect of real-time capabilities, which is crucial for any AVT system. There are only a handful of instance segmentation methods specifically designed for real-time usage [11]–[13]. However, there are no publicly available experimental results regarding pedestrian segmentation in street scenes, such as on the well-known Cityscapes urban street dataset [14], using the before-mentioned real-time instance segmentation methods.

Real-world datasets are usually relatively small in size due to the difficult and time-consuming manual annotation procedure. Hence, the training of real-time instance segmentation models on a real-world dataset might cause overfitting or could yield mediocre if not bad results. On the other hand, synthetic datasets are larger because the generation of synthetic data is usually cheap and quick, since no manual annotation is involved. For example, the real-world urban street scene Cityscapes dataset has only 5000 annotated images, while the synthetic urban street scene Synscapes [15] dataset contains 25000 annotated images. Furthermore, state-of-the-art GPU accelerated image rendering software suites [16], [17] are able to generate photorealistic images of high quality. Such that it is rather difficult to distinguish synthetic images from real-world images, as shown in figure 1. Additionally, the advantage of these highly photorealistic images is that they have a smaller gap in the difference of the data distribution w.r.t. real-world data.

Therefore, we propose a transfer learning-based training approach to improve the overall accuracy of real-time instance segmentation models for pedestrians by leveraging the large amount of data available in the synthetic dataset [15] and the correct data distribution of the real-world dataset [14]. First, we train the state-of-the-art real-time instance segmentation

*This result is part of a project that has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreement No 870133.
models on the large-scale synthetic Synscape dataset. The relatively large amount of data prevents the real-time instance segmentation models from overfitting, which might be the case for using only small-scale real-world datasets. Secondly, we use the weights trained on synthetic data and fine-tune the real-time instance segmentation networks on the real-world Cityscapes dataset using different amounts of training data. Here, we wanted to see how many real-world data samples are required to outperform the real-time instance segmentation models trained solely on the entire real-world dataset.

The main contribution of our work can be summarized as the following:

- We proposed a novel training methodology based on transfer learning for real-time instance segmentation methods for pedestrians,
- A comparison of the proposed transfer learning-based training approach by using different amounts of real-world data for fine-tuning,
- Extensive experiments of three state-of-the-art real-time instance segmentation methods on the Synscape synthetic dataset and on the Cityscapes real-world dataset.

II. RELATED WORK

A. Instance Segmentation

One of the first instance segmentation method that achieved greater success and served as a base for many other works is the Mask R-CNN [6]. The Mask R-CNN network extended its predecessor, the Faster R-CNN model [18], by adding a branch to perform pixel-wise segmentation in the predicted bounding boxes. Therefore, most of the state-of-the-art instance segmentation models are two stage methods which follow the Mask R-CNN [6] strategy. They first detect the objects and then use the bounding boxes to predict the segmentation mask [9], [10]. Hence, these methods tend to have a higher runtime, which makes them useless for safety-critical real-time applications such as pedestrian segmentation for AVT systems.

B. Real-Time Instance Segmentation

YOLACT [19] and YOLACT++ [11] is the first instance segmentation method to achieve real-time. This was achieved by performing in parallel the generation of a set of prototype masks and predicting per-instance mask coefficients. In YOLACT++ the authors increased the performance over their first method, YOLACT, by leveraging deformable convolutions in the backbone to get better feature sampling and to achieve higher robustness against scale and rotation variations. BlendMask [12], on the other hand, leverages the combination of the top-down and the bottom-up approaches. The prediction of the final segmentation masks is achieved by combining the set of base masks together with the attention masks. Finally, SipMask [13] subdivides a bounding box into subregions and introduces a spatial preservation module to retain instance-level spatial information. Compared to YOLACT, which uses only one set of coefficients for the prototype masks, SipMask requires a set of coefficients for every subdivided region within a bounding box.

C. Pedestrian Instance Segmentation

As mentioned before, most of the instance segmentation methods are based on the Mask R-CNN strategy. The same is valid for pedestrian instance segmentation approaches. The authors of [20] extend the original Mask R-CNN by including prior knowledge about the body parts’ proportion of pedestrians. This enables their Part Mask R-CNN [20] model to learn more information about pedestrian instances. The method proposed in [21] focuses on occluded pedestrian instances and enhances the segmentation result by leveraging pedestrian count and proposal similarity information. However, these methods have the same limitation of not being real-time capable.

III. METHODOLOGY

Our transfer learning approach for real-time pedestrian instance segmentation leverages two important properties from the synthetic Synscapes and real-world Cityscapes datasets: 1) the large amount of data in Synscape in order to prevent overfitting, and 2) the real-world data distribution of Cityscapes to achieve better performance in a real-world scenario. Hence, our training methodology can be briefly summarized in two steps:

1) Pretraining of an instance segmentation method on the photorealistic synthetic Synscapes dataset.
2) Fine-tuning on the Cityscapes dataset using different amounts of training data samples.

A. Training Approach

The introduced transfer learning-based training pipeline for real-time instance segmentation of pedestrians is shown in figure 2. First, we pre-train the real-time instance segmentation methods, i.e. YOLACT++ [11], SipMask [13], and BlendMask
Fig. 3: Our presented transfer learning approach. First, the baseline instance segmentation method is trained using the photorealistic synthetic Synscape [15] dataset. We add to the model name the prefix "S" to indicate that it was trained on a synthetic dataset. Then, we finetune the instance segmentation model trained on synthetic data with the real-world Cityscapes [14] dataset. We add to the final model name the prefix "S+R" to indicate that it was pretrained on synthetic data and finetuned on real-world data.

[12], on the Synscape synthetic dataset using only the pedestrian class. The next step in our training pipeline consists of using the pre-trained model weights from the training on the synthetic dataset and fine-tuning them using the real-world dataset. Moreover, we investigated the dependency of the performance of the fine-tuned models w.r.t. the number of real-world data samples were used for training. Therefore, we calculated the Cityscapes training set statistics regarding the pedestrian instance distribution, as depicted in figure 2. The pedestrian instance distribution gives the information about the number of images w.r.t. the number of pedestrian instances. For example, it is interesting to observe that there are more than 600 images which contain zero pedestrian instances. Moreover, the amount of images with less than three pedestrian instances is relatively high. Therefore, we decided to filter out all the images from the training dataset, which contained less than three pedestrian instances. This filtering step equalizes the pedestrian instance distribution better throughout the training set. After applying this filtering step, the training set has approximately 1500 images left. Hence, we chose to fine-tune the real-time instance segmentation models, pre-trained on the Synscapes dataset, using 50, 500, 1000, and 1500 randomly selected training data samples.

B. Implementation Details

We evaluated transfer learning-based training pipeline for real-time instance segmentation of pedestrians using the original repositories of YOLACT++ [11], SipMask [13], and BlendMask [12]. Moreover, we followed the author’s recommendation regarding the hyperparameter settings. We trained YOLACT++ with SGD using a learning rate of 0.001, a weight decay of 0.0005, and a momentum of 0.9. SipMask and BlendMask were trained with SGD using a learning rate of 0.01, while the other hyperparameters were the same as for YOLACT++. The batch sizes for YOLACT++ and BlendMask were set to 16, while for SipMask only a batch size of 8 fitted to the memory. All three methods employed the ImageNet pre-trained ResNet-50 [22] backbone. We trained all the methods for about 75000 iterations on our Workstation PC with an NVIDIA Quadro GV100 GPU and Ubuntu 20.04 LTS.

C. Evaluation Metrics

The metric used for the evaluation of the different baselines and fine-tuned models is the well-known average precision (AP), which was introduced by [23]. Generally speaking, the AP can be defined as finding the area under the precision-recall curve:

\[ AP = \int_{0}^{1} p(r)dr \]

where \( p \) and \( r \) denote the precision and recall, respectively. The precision and recall are defined as:

\[ p = \frac{TP}{TP + FP} \]

\[ r = \frac{TP}{TP + FN} \]

where \( TP \) denotes the true positives, \( FP \) the false positives, and \( FN \) the false negatives. An instance segmentation mask prediction is considered to be TP if the intersection over union (IoU) is greater than a given threshold. The IoU can be simply defined as the calculation of the overlap between the predicted mask and its ground truth pair:

\[ IoU = \frac{|R_p \cap R_g|}{|R_p \cup R_g|} \]

where \( R_p \) and \( R_g \) denote the predicted and ground truth mask region, respectively. We follow previous works [11]–[13], where a common way of evaluating instance segmentation methods is to use the overall AP, the AP at \( IoU > 0.5 \) (AP\(_{50}\)), and the AP at \( IoU > 0.75 \) (AP\(_{75}\)).

IV. Evaluation

In this section, we begin with a detailed description of the used datasets. Then, we provide a quantitative and qualitative comparison of the real-time instance segmentation baselines w.r.t. the models trained using our proposed transfer learning-based training strategy using different amount of training data samples during fine-tuning.
A. Datasets

All experiments were conducted using the real-world Cityscapes [14] dataset and the synthetic Synscapes [15] dataset. Both datasets contain images of urban street scenes and are among the most relevant datasets to be used for AVT development. The Cityscapes dataset was created in cooperation with the Daimler AG RD Department. The images were captured in 50 cities located mostly in Germany. The Cityscapes dataset contains 5000 fine and 20000 coarse annotated images. We used the 5000 fine annotated images in our experiments since only the fine annotations are suitable for the instance segmentation task. The Synscapes dataset contains 25000 photorealistic images. The dataset did not generate the images by using a driven path through the virtual environment. Instead, the authors from Synscapes [15] created a unique scene for each image, which tremendously increases the variety of the data distribution. For our experiments with the Synscapes dataset, we used a split of 24000 data samples for training and 1000 for validation.

B. Quantitative Evaluation

Table I shows the comparison of the baselines trained solely on the Cityscapes training set or the synthetic Synscapes dataset w.r.t. to the models trained with our training strategy. We can observe that models of YOLACT++ and BlendMask trained on the Cityscapes dataset perform better w.r.t. the models trained solely on the synthetic Synscapes dataset. However, one interesting observation to make, is that the SipMask model trained on the synthetic dataset performs better by a large margin, compared to the model trained on the real-world dataset. The reason for such a large discrepancy is most probably due to the relatively larger network model. Hence, the SipMask model requires significantly more training data to increase its performance.

On the other hand, it is surprising to observe that for all of the used real-time instance segmentation models trained on synthetic data in the first training step, fine-tuning on only 50 real-world images from the Cityscapes dataset is enough to be on par or outperform the models trained solely on the entire Cityscapes dataset. Moreover, we see a significant performance boost for the SipMask model pretrained on the Synscapes dataset and fine-tuned on 50 real-world images from the Cityscapes dataset. The reason for this high increase in performance can be explained by the larger amount of training data used in the first training step of our presented transfer learning-based training strategy.

Furthermore, it is evident that the best models of our proposed training strategy approach outperform by a large margin the baselines trained solely on the relatively small real-world Cityscapes dataset. Hence, the YOLACT++ model trained using our approach increases the overall AP by +2.6% compared w.r.t. to its baseline. Further, the transfer learning-based training strategy applied on the SipMask model boosts its performance by a large margin of +9.9%. Finally, our training strategy improves the BlendMask model by +4.1% w.r.t. to its baseline trained only on real-world data.

Finally, we report the inference time of YOLACT++, SipMask, and BlendMask in Table II. It is clear that all the methods are capable of real-time inference under the defined settings.

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Input size</th>
<th>Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLACT++*</td>
<td>ResNet-50</td>
<td>640 × 480</td>
<td>34.5</td>
</tr>
<tr>
<td>SipMask*</td>
<td>ResNet-50</td>
<td>640 × 480</td>
<td>33.8</td>
</tr>
<tr>
<td>BlendMask*</td>
<td>ResNet-50</td>
<td>640 × 480</td>
<td>36.7</td>
</tr>
</tbody>
</table>

* The inference time is valid for all the different trained models.
C. Qualitative Evaluation

The qualitative comparison of the real-time instance segmentation models trained with our training strategy and their baselines is shown in figure 4. Again, it is clearly visible that our presented transfer learning-based training pipeline visually achieves the best results. Moreover, if we take a detailed look at the qualitative results, we can observe that the instance mask predictions are more detailed for the models which were obtained with our training pipeline. For example, the models trained solely on real-world data tend to include the space between the legs of a person in the mask prediction. Contrary, the methods trained with our two-stage training approach manage to correctly predict the space between the legs of a person as background. We can assign this to the synthetic data pre-training step, because synthetic datasets have perfect annotations. Unlike the perfect annotations in synthetic data, real-world datasets suffer from faulty ground truth segmentation masks due to human error during the annotation process. Finally, we can state that our transfer learning-based training strategy mitigates the drawbacks of solely using relatively small-scale real-world datasets and increases the qualitative performance of the used real-time instance segmentation methods by a large margin.

V. DISCUSSION AND FUTURE WORK

The introduced transfer learning training strategy combining synthetic and real-world data helps to boost the performance of real-time instance segmentation methods on pedestrians. Furthermore, it is interesting to observe that even a small amount of real-world data is sufficient to fine-tune the models trained on synthetic data to achieve on par or even better results. We believe that the presented approach could help for less investigated use cases, such as manufacturing or assembly processes, where mostly no public datasets are available. To emphasize our premise, we applied the best performing model, i.e., BlendMask, to a robotic workcell use case for human intrusion detection, as shown in figure 5. Even though the model never saw such data during training, it is able to segment the human entering the robot’s workspace remarkably well.

However, the presented approach still requires time-consuming manual annotation of real-world data. To mitigate this issue, we plan to investigate unsupervised domain adaptation (UDA) methods, such as [24], which brings the learned source data distribution closer to the unlabeled target data distribution.
VI. CONCLUSION

In this work, we introduced a simple yet effective transfer learning-based training approach for real-time instance segmentation of pedestrians. The core concept of the introduced approach was to leverage the advantages of large-scale synthetic and real-world datasets. Therefore, we investigated the use of a two-stage training pipeline, where the models were trained using the synthetic Synscapes dataset in the first step and then fine-tuned using the real-world Cityscapes dataset. The first training step provided good weight distribution due to a large amount of available training data samples in the synthetic dataset. Additionally, this prevented the models from overfitting. These pre-trained weights were then used in the second training step on real-world data. Moreover, the conducted experiments showed that only a small amount of real-world data was necessary for fine-tuning the models and outperforming their baselines.

Our work showcases the advantages of a combined training approach of synthetic and real-world datasets:

• With the proposed training approach, the amount of required real-world data can be reduced, and thus the effort of creating such a dataset based on manual annotations.
• Big neural network architectures that require a large amount of training data can be successfully trained without overfitting by using our approach.
• The introduced approach visually improves the mask prediction by leveraging the fine-grained mask annotations from synthetic data.

Finally, our transfer learning-based training strategy can help in the development of safety-critical features in automated driving systems, as it showcases a large performance boost on real-time instance segmentation of pedestrians.

REFERENCES

An Improved YOLOv5 Real-time Detection Method for Aircraft Target Detection

Yuxi Chen, Jiaye Yang, Jianfeng Wang, Xue Zhou, Jianxiao Zou, Yun Li
Shenzhen Institute of Advanced Study
University of Electronic Science and Technology of China
Shenzhen, China

{imcyx, jiaye.yang}@std.uestc.edu.cn, zhouxue@uestc.edu.cn, Yun.Li@ieee.org

Abstract—To address the lack of accuracy and speed of aircraft detection in complex background images, the YOLOv5 model is improved in this paper to meet the growing demand for aircraft detection. Firstly, Mosaic-9 data enhancement is performed on the dataset, and MobileNet V3 Small is used to replace ResNet feature extraction network to enrich small target samples and improve feature extraction speed. Secondly, over-fitting is prevented by Label Smoothing, and channel pruning is used to reduce the complexity and redundancy of the neural network. Finally, TensorRT is used to optimize the underlying hardware resources to improve the detection speed. The experiments use Military Aircraft Detection Dataset for model training and validation on NVIDIA Tesla P100 16GB GPU. The results show that our improvements maintain the accuracy of the model while effectively increasing the training speed and detection speed of the model. Our codes are available at https://github.com/imcyx/Military-Detect

Index Terms—YOLOv5, Aircraft detection, Data enhancement, Model acceleration

I. INTRODUCTION

Aircraft play an irreplaceable role in both the civil and military sectors, and it has become important to detect aircraft in images. However, there is a great deal of uncertainty in the capture of aircraft images, with differences in the angle of the image and the environment in which it is taken. In addition, aircraft of approximately the same appearance and shape may belong to completely different categories; and aircraft of the same type may have completely different graffiti, all of which can result in a reduction in recognition accuracy.

Most of the traditional aircraft detection methods are manual feature extraction. A large number of image pre-processing operations, such as mean filtering and histogram equalization, are performed before object detection, mostly relying on low-level information of the image. With the rise of deep learning technology, this end-to-end learning approach can automatically extract image features, and different depths of network layers can extract different levels of image information. The shallow layer network can extract texture and shape information of the image, and the deep layer network can obtain semantic information, which brings great benefits to the detection and classification of aircraft targets.

As Figure 1 shows, YOLOv5 target detection network is used as our base model, and some improvements are proposed to increase the detection accuracy and speed in this paper:

1. We use the Mosaic-9 data enhancement method to expand the original dataset images. By randomly cutting, randomly scaling, and randomly arranging 9 images, a new image is formed as the input to the neural network. A lot of new image backgrounds are generated by data enhancement, which enhances the robustness and accuracy of aircraft detection.

2. We replace the previous CSPDarkNet with the lightweight Mobilenet V3 small[1] as the new backbone network, which can greatly reduce the model complexity without reducing the network accuracy.

3. We prune the channels of deep convolutional neural network (CNN). Channel pruning was proposed in [2] and can reduce the size of the model, reduce the memory usage at runtime, and reduce the number of computational operations without affecting the accuracy.

4. We use the introduction of label smoothing method in the prediction layer of YOLOv5. The label smoothing method will help the model to train around mislabeled data, thus improving its robustness and performance.

5. The Yolo-trained model is sent to TensorRT to optimize the engine and then applied to GPU inference, further reducing the burden on the model and greatly improving the model
inference speed with an acceptable loss of accuracy.

With the above five improvements, the final model has greatly improved in accuracy and detection speed compared to the previous YOLOv5 base network. In particular, the detection speed can already reach the standard of real-time detection.

II. RELATED WORK

A. YOLO detection models.

The YOLO [3] algorithm is proposed by Redom J et al. It also belongs to the one-stage target detection algorithm, which mainly uses a single neural network to predict the location of the bounding box and the class of objects inside the box, so it runs very fast, and one of the fast versions can even reach 155 frames per second. YOLO9000 [4] removes the fully connected layer from the YOLO algorithm. And the YOLOv3 [5] replaces the loss function in YOLO9000 and uses Darknet-53 as the feature extraction network, which improves the detection accuracy while guaranteeing the detection speed. Although the YOLOv3 algorithm has the advantages of fast detection speed, low background false detection rate, and strong generalization ability, however, the YOLOv3 algorithm still has shortcomings, such as low detection accuracy, high influence by environment such as illumination, and poor detection for small-scale targets. To address these problems, Hurtik [6] et al. proposed a Poly-YOLO algorithm based on YOLOv3 using a lightweight SE-Darknet-53 feature extraction network, which improved the mAP by 40% and increased the detection speed by a factor of two, however, the algorithm is less effective in detecting overlapping targets. A comparison of these early methods is shown in Tabel I.

### TABLE I

<table>
<thead>
<tr>
<th>Model</th>
<th>Speed(FPS)</th>
<th>Accuracy(mAP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Faster R-CNN</td>
<td>7</td>
<td>59.1</td>
</tr>
<tr>
<td>YOLOv2</td>
<td>67</td>
<td>44.0</td>
</tr>
<tr>
<td>SSD</td>
<td>46</td>
<td>50.4</td>
</tr>
<tr>
<td>YOLOv3</td>
<td>45</td>
<td>57.9</td>
</tr>
</tbody>
</table>

It can be seen that starting from YOLOv3, the YOLO method can already approach the traditional two-stage methods. As a further improvement, Bochkovskiy et al. [8] proposed the YOLOv4 algorithm, mainly intended for industrial applications, with a lot of balance between speed and accuracy. In the same year, Jocher proposed YOLOv5, which can easily convert Pytorch trained weight files into mobile-ready onnx models that can be directly deployed for applications on mobile devices, and is currently the best target detection network for inference. The backbone network of YOLOv5 is shown as Figure 2.

![Fig. 2. Yolov5’s Backbone removes the Focus module (which facilitates the export of models for deployment) and the Backbone consists mainly of CBL, Bottleneck CSP/C3 and SPP/SPPF as shown in the diagram.](image)

B. Aircraft detection methods

The current methods for aircraft detection mainly include two directions, one is based on traditional image processing and the other is based on deep learning.

1) Traditional methods: Detection algorithms based on image processing and traditional classifiers process as follows: firstly, image pre-processing, secondly, intercepting the target part from many backgrounds by target segmentation, then extracting the image feature points, and finally, inputting the feature points to the classifier. For example, Camps et al. [9] proposed JPEG2000 classifier based on a scalable feature extraction criterion from coarse to fine, which uses integral convolution to learn a collection of random trees to perform multiple classification tasks on JPEG2000 image database. sumalatha et al. [10] based on local subwindow extraction technique and random decision tree to operate directly on pixel values without feature extraction. feature extraction is required. Ghaith et al. [11] first transformed the features and used fuzzy clustering to achieve aircraft image classification based on the similarity between the features. Zhu et al. [12] improved on the BoW classification algorithm by optimizing its parameters. Molchanov et al. [13] fused the extracted Doppler features as the feature volume for image classification.

2) Deep learning methods: Cai et al. [14] proposed a resolution-aware CNN framework and added a super-resolution layer within the framework to significantly improve the classification of low-resolution images. Mac et al. [15] proposed an efficient spatio-temporal prior model to estimate the probability of occurrence of a given object class at a given location conditional on the location and the time of capture. Ge et al. [16] built a complementary model in a weakly supervised manner to retrieve dominant objects detected by convolutional neural networks, and accomplish the instance segmentation task by using Mask R-CNN when only image-level labels are given. Qi et al. [17] first considered the spatial relationship between objects to construct a discriminative representation of fine-grained images, and proposed an effective strategy for selecting spatial relationships.
III. METHODOLOGY

A. Mosaic-9 data enhancement

The main idea of Mosaic is to randomly crop four images and stitch them onto a single image as new training data. The background of the images is greatly enriched by the Mosaic data enhancement, which enables the network to better handle the bias caused by the different shooting environments and shooting angles.

In this experiment, we used Mosaic-9, an enhanced version of the Mosaic method, to combine nine images into one image by random cropping, random scaling and random alignment, the details of which are shown in Figure 3. This new image is then passed into the neural network to learn. This is equivalent to passing in 9 images for learning at once, and the normalization process is to calculate the average of the 9 images. After the new images are stitched together, some of the images and detection frames are overwritten by neighboring images. When the frame of a picture is beyond the edge between two pictures, the part of the frame or picture that is beyond the split line needs to be removed.

B. MobileNet V3

MobileNet has no residual structure and is a straightforward and streamlined architecture. It uses depthwise separable convolution to improve the computational speed of the network, and depthwise separate convolution includes depthwise convolution and pointwise convolution, and introduces Width Multiplier is introduced to adjust the number of channels of convolution output to balance the computational speed and accuracy of the network more easily.

The MobileNet V3 model we use has made improvements in the following aspects:

1. The SE module is introduced. By explicitly modeling the interdependencies between the convolutional feature channels of the network, it guides the weight update and enhances the ability of the network to extract features.

2. Modify the tail structure. The 1x1 convolution before the average pooling layer is placed after the pooling layer, because the 1x1 convolution will occupy a lot of computing time, and placing it after the pooling layer can reduce the amount of computation.

3. Modify the number of channels. The number of channels in the first convolution kernel is modified from 32 to 16 to further reduce the model complexity.

4. Replace the loss function. Replace the ReLU activation function with the swish nonlinear activation function to improve the network accuracy, which is described by the following equation:

\[ swish(x) = x \cdot \sigma(x), \]  \hspace{1cm} (1)

where \( \sigma \) representations the sigmoid function. However, the sigmoid function can greatly increase the computational cost. The ReLU6 function has been implemented in many hardware and software frameworks, is easy to quantize and deploy, performs better even with 16-bit floating point or 8-bit integer low-precision operations, and is fast in computational reasoning. Therefore, we use the h-swish activation function to approximate the swish function with the following equation:

\[ h_{-swish}(x) = \frac{x \cdot (ReLU6(x + 3))}{6}. \] \hspace{1cm} (2)

The comparison graph of swish and h-swish activation functions is shown in Figure 4. It is experimentally verified that using the h-swish activation function does not reduce the network accuracy, but can improve the network inference speed.

C. Channel pruning

The superior performance of CNNs usually comes from millions of trainable parameters, and the high computational cost makes their deployment difficult on the ground. Moreover, CNNs require even more space for intermediate activation and response storage than for model parameters during inference, as well as longer computation time due to computationally intensive convolutional operations on high-resolution images.

The model channel pruning schematic is shown in Figure 5. Channel pruning reduces the model complexity by forcing channel-level sparsity. It can be directly applied to CNN architectures, introduces minimal overhead in the training process, and generates models that do not require special hardware and software gas pedals. Channel pruning can reduce the model size, reduce the runtime memory footprint, and reduce the number of computational operations without affecting
Fig. 5. Pruning is one of the ways to improve model efficiency by efficiently generating models that are smaller, have better memory utilisation and are faster in the forward direction. In our work, we first perform sparse training by adding $L_1$ parametrization to the output of the BN layer, and generate the weight matrix as a sparse matrix by a back-propagation algorithm. Then, we prune according to the pruning ratio $\gamma$ by sorting the weights from smallest to largest. Finally, we fine-tune the pruning results to obtain the final new network.

the accuracy, making it widely used in practical scenarios. Current pruning algorithms are divided into structured and unstructured pruning. The unstructured pruning method requires quantization and encoding of sparsely connected networks to reduce the actual storage space of the model, and requires specialized hardware devices and computation methods to achieve model inference acceleration. Therefore, structured pruning is used in this paper. Channel pruning screens some redundant connections existing in the neural network in the way of model reconstruction, and these structures contribute little to the model performance. Removing this part of neurons can effectively reduce the model complexity, while almost not affecting the accuracy of the network, and even improving the comprehensive performance of the network. The calculation formula is as follows:

$$
\hat{z} = \frac{z_{in} - \mu_B}{\sqrt{\sigma_B^2} + \epsilon},
$$

(3)

$$
z_{out} = \gamma \hat{z} + \beta,
$$

(4)

where $\mu_B$ and $\sigma_B^2$ are the mean and variance of a batch of input data, $\gamma$ and $\beta$ are pre-defined parameters. The speed of network convergence is controlled by setting different initial values, which also have an impact on the accuracy.

D. Label Smoothing

Label Smoothing is first used in classification algorithms and later introduced in target detection algorithms. Essentially Label Smoothing is a regularization method that helps the model to train around mislabeled data, thus improving the robustness and detection accuracy of the model. Label Smoothing prevents the model from falling to the overfitting stage by reducing the confidence of easy samples and increasing the confidence of difficult examples, which is expressed by the equation:

$$
q'_i = (1 - \epsilon)q_i + \frac{\epsilon}{K},
$$

(5)

where $q_i$ denotes the true label, $\epsilon$ is a very small constant, and $K$ represents the number of categories for classification. After Label Smoothing, it can reduce the problem of over-reliance on labels and improve the accuracy of labels.

E. TensorRT Optimization

TensorRT is an SDK based on the Nvidia CUDA programming model and is primarily used for high-performance deep learning inference. This specific optimization provides api and parser to import the trained deep learning model, and after weight and activation accuracy calibration, layer and tensor fusion, kernel auto-tuning, dynamic tensor display and multi-stream execution finally generates an optimized runtime engine that can be deployed in various environments. The optimization flowchart is shown as Figure 6.

In this paper, to further lighten the YOLO-trained model, we feed the model into TensorRT to optimize the generated Engine engine before applying it to GPU inference. In the final GPU inference phase, the optimized engine is deserialized and parsed, and the input data is copied from the CPU to the GPU when the inference request is issued, and the result is returned to the CPU asynchronously when the inference is completed.

IV. EXPERIMENTS

A. Dataset

Our experiments are based on the Military Aircraft Detection Dataset \(^1\), which contains 40 species and 6178 military aircraft images, including bounding box in PASCAL VOC format (xmin, ymin, xmax, ymax) and popular aircraft types from China, USA, Russia, and Europe.

B. Experiment details

Our experiments were done in a Linux-based environment, specifically Ubuntu-18.04, and the code is based on python version 3.6, using Pytorch 1.11.0 and other third-party libraries

\(^1\)https://www.kaggle.com/a2015003713/militaryaircraftdetectiondataset
Fig. 7. The final convergence function of loss for 50 epochs based on the improved YoloV5 model is shown here. As can be seen, with iterations of training, mAP@0.5 eventually stabilises at around 0.6. It is worth noting that there is some jitter in the curve due to the multiple breakpoint training performed, but this does not influence our final experimental results.

used in the experiments. During the acceleration phase, we training with Cuda 11.4 and TensorRT 8.0.3.4.

Furthermore, based on hardware conditions, setting the training dataset to 75%, the validation dataset to 20% and the test dataset to 5% is Reasonable. We set the initial learning rate as 5e-2 and the final OneCycleLR learning rate as 1e-2 to Prevent over-fitting. The total training batch epoch=50, batch-size=2, and the loss function is shown in Figure 7.

C. Experiment results

<table>
<thead>
<tr>
<th>Model</th>
<th>mAP@0.5</th>
<th>mAP@0.5:0.95</th>
<th>FPS</th>
</tr>
</thead>
<tbody>
<tr>
<td>YOLOv5</td>
<td>62.1</td>
<td>53.2</td>
<td>84</td>
</tr>
<tr>
<td>YOLOv5-MobileNet V3</td>
<td>60.1</td>
<td>51.6</td>
<td>120</td>
</tr>
<tr>
<td>YOLOv5-Channel pruning</td>
<td>60.8</td>
<td>51.9</td>
<td>99</td>
</tr>
<tr>
<td>YOLOv5-Label smoothing</td>
<td>63.5</td>
<td>54.3</td>
<td>86</td>
</tr>
<tr>
<td>YOLOv5-TensorRT</td>
<td>58.6</td>
<td>49.1</td>
<td>480</td>
</tr>
<tr>
<td>Ours</td>
<td>59.3</td>
<td>49.8</td>
<td>555</td>
</tr>
</tbody>
</table>

It can be observed from the Table II that several of our improvements to YOLOv5 have resulted in speedups. In particular, the use of TensorRT improves the FPS by nearly 7 times compared to the original YOLOv5 network. However, pursuing speed too much will inevitably also bring about a decrease in detection accuracy, so we use the Label smoothing strategy to balance accuracy to some extent.

Some of the experimental results of our model are shown in Tabel II. After 50 iterations, the final loss and mAP values of the model are shown in Figure 7. From the figure, we can see that the loss of the model is gradually decreasing and the model is gradually converging. And the final mAP value, recall, and accuracy of the model are approaching 60%, which proves that the accuracy of the optimized model does not drop much.

The final model after TensorRT acceleration has a great improvement in detection speed, although it loses some accuracy. The detection results of model optimization using FP16 accuracy are shown in Figure 8.

V. Conclusion

In this paper, we make a number of improvements to YOLOv5 for the specific task of aircraft detection, with the main goal of improving the inference speed of the model as much as possible with essentially no loss of model accuracy. Specifically, we use the Mosaic-9 data enhancement method to enhance the model generalization performance. In addition, we adopt MobileNet V3 Small instead of CSPDarkNet as the feature extraction network and further reduce the model complexity by channel pruning. Label Smoothing is used as a regularization means to compute the loss. Finally, the GPU hardware resources are optimized by TensorRT to substantially improve the model processing speed. Experiments show that the FPS of our final model is more than ten times faster than before.
ACKNOWLEDGMENT

This work is supported in part by the Guangdong Natural Science Foundation under Grant 2021B1515120078.

REFERENCES


A Unified System for Encryption and Multi-Secret Image Sharing Using S-box and CRT


*Nanoelectronics Integrated Systems Center (NISC), Nile University, Giza 12588, Egypt.
†Engineering Mathematics Department, Faculty of Engineering, Cairo University, Giza, 12613, Egypt.
‡School of Engineering and Applied Sciences, Nile University, Giza, 12588, Egypt.
agradwan@ieee.org

Abstract—Multi-Secret Image Sharing (MSIS) is used when multiple images need to be shared to multiple participants, but the images can not be recovered without the presence of all shares. In this paper, a unified system for performing encryption and (n,n)-MSIS is proposed. While MSIS is based on the XOR operation, encryption combines the utilization of Chinese Remainder Theorem (CRT), SHA-256, and S-box for improved security. The same designed system is used for the generation of secret shares and the recovery of secret images. In addition, a sensitive system key is designed where three pairwise relatively prime subkeys are automatically generated for utilization in the CRT. The resulting secret shares pass statistical evaluation criteria such as RMSE, correlation, and entropy, and give good results for differential attack measures, and runtime. In addition, the proposed system succeeds in passing the NIST SP-800-22 statistical test suite and key sensitivity measures.

Index Terms—CRT, Image encryption, MSIS, S-box, SHA-256

I. INTRODUCTION

In modern telecommunication, digital data is important and huge amounts of images are stored and transferred through the internet. This increased the awareness of privacy and internet security, making the protection of digital images an important requirement. Consequently, many research efforts are performed in information security fields like steganography, watermarking, cryptography, and Secret Sharing (SS). SS was proposed by Shamir [1] and Blakley [2] with (k, n)-threshold, where a dealer shares a secret into n shares and distributes them to n participants. The secret can only be recovered by k or more participants, where less than k participants can not get a clue about the secret. Based on their work, the idea of Secret Image Sharing (SIS) was then proposed by Thien and Lien [3], where an image is shared as n image shares, where each share has a size equals 1/k of the original image size. Furthermore, Lin and Tsai [4] proposed the idea of meaningful shares by using steganography techniques to hide shares in meaningful images, since noise-like shares would attract attention.

In addition, some recent image encryption literature focuses on encrypting multiple images at once, where Wang et al. [5] used 2D chaotic system with 3D Discrete Cosine Transform (DCT) to encrypt three grayscale images. Ye et al. [6] also used DCT to make visually meaningful multiple images encryption. The interest in multiple secret sharing also occurs in the recent literature introducing Multi-Secret Image Sharing (MSIS). In MSIS, the most common methods for achieving secret sharing are XOR and Chinese Remainder Theorem (CRT) [7]. Deshmukh et al. [8] proposed a (n,n)-MSIS system based on the XOR and used CRT only as encryption, which was not secure enough and had a problem with even number of secret images. Prasetyo et al. [9] proposed a (n,n)-MSIS system based on the XOR and used CRT and added a simple chaotic map to improve security. They also proposed three methods to solve the problem of even number of secret images. Prasetyo and Hesia [10] added generalized chaotic image scrambling together with the CRT and XOR to improve the security of their proposed scheme. Prasetyo and Guo [11] mainly concentrated on the problem of even number of secret shares presented in [8] and also added hyperchaotic scrambling to increase system security. Guo et al. [12] used Beta chaotic map for performing confusion and diffusion on the plain secret images. Most of the previous MSIS schemes, which are based on [8], proposed solutions for the problem of even number of input with minor modifications to increase the system security but had a problem with the keys used in the CRT as they were too small, which is a major security issue.

This paper contributions include proposing a new MSIS system with secure encryption, where the same system is used in both generation and recovery. The proposed system combines encryption and MSIS using XOR, and improves the security by the utilization of CRT, S-box and Secure Hash Algorithm-256 (SHA-256) [13]. The proposed system also addresses the system key issues found in recent schemes by designing a sensitive system key that generates long subkeys for CRT, which can be long enough to resist brute force attacks. Evaluation of the system, with respect to security and performance, demonstrates good results that are either comparable to or better than related schemes.

The paper is organized as follows. Section II presents the proposed (n,n)-MSIS system along with the proposed subkeys generation method. Section III explains the experiments, results, and comparisons with previous schemes. Finally, Section IV briefly gives the conclusions.

This paper is based upon work supported by Science, Technology, and Innovation Funding Authority (STIFA) under grant number (∗#45631).
II. PROPOSED SYSTEM

The proposed system is the same for generation and recovery as shown in Fig. 1. For the generation of secret shares, the $n$ secret images $\{I_1, I_2, \ldots, I_n\}$ are, first, XORed together to form the image $T_1$ as follows:

$$T_1 = I_1 \oplus I_2 \oplus \cdots \oplus I_n.$$  \hspace{1cm} (1)

The image $T_1$ is passed to SHA-256, which produces a 256-bit digest (32 bytes). The digest bytes are then XORed with the pixels of the image $T_1$, where the first byte from the digest is XORed with the three channels of the first pixel, then the next byte is XORed with the next pixel’s channels and so on. A round-robin approach is used, in which the same 32 bytes digest is used repeatedly with each sequence of consecutive 32 pixels to get the image $T_2$. By using SHA-256, the image $T_2$ is sensitive to changes in the image $T_1$ and, consequently, to any slight change in the input secret images. $T_2$ is then passed to the CRT module to encrypt it by solving the following system of congruences for each pixel using three pairwise relatively prime subkeys [7]:

$$x \equiv R \mod k_1,$$

$$x \equiv G \mod k_2,$$

$$x \equiv B \mod k_3,$$

where $R, G, B$ are the three channels’ values, $k_1, k_2$, and $k_3$ are the subkeys generated from the system key $K$, and $x$ is the resulting encrypted value in the range $[0, k_1k_2k_3]$. In order to limit the size of each pixel in any channel to 8 bits, $x$ is reduced $\mod 256$. The reduced $x$ for each pixel is then saved in the three channels to form the image $T_3$. Since $T_3$ has identical values in the three channels of each pixel, Rijndael S-box of the Advanced Encryption Standard (AES) [14] is used to perturb this similarity and randomly substitute one of the three channels. The multiplexer performs the operation $x \mod 3$ for each pixel to get one of the values $\{0, 1, 2\}$ that decides which channel will be substituted using the S-box. After the S-box, each pixel is XORed with the previously encrypted pixel using delay (cumulative XOR) [15] to get the encrypted image $T_4$. $T_4$ is finally XORed with each secret image to generate the secret shares $\{S_1, S_2, \ldots, S_n\}$ as follows:

$$S_i = I_i \oplus T_4 \text{ for } i = 1, 2, \ldots, n.$$  \hspace{1cm} (3)

The recovery scheme is the same as the generation scheme because XORing the secret shares together yields the same $T_1$ for even values of $n$ as follows:

$$T_1 = S_1 \oplus S_2 \oplus S_3 \oplus S_4,$$

$$= I_1 \oplus T_4 \oplus I_2 \oplus T_4 \oplus I_3 \oplus T_4 \oplus I_4 \oplus T_4,$$

$$= I_1 \oplus I_2 \oplus I_3 \oplus I_4.$$  \hspace{1cm} (4a) \hspace{1cm} (4b) \hspace{1cm} (4c)

$T_4$ goes into the same previously described steps to get the same $T_4$ which is XORed with the secret shares to get the recovered secret images $\{R_1, R_2, \ldots, R_n\}$ as follows:

$$R_i = S_i \oplus T_4,$$

$$= I_i \oplus T_4 \oplus T_4 = I_i, \text{ for } i = 1, 2, \ldots, n.$$  \hspace{1cm} (5a) \hspace{1cm} (5b)
In the case of odd values of \( n \), the modification presented in [11] can be adopted.

In order for the CRT to work properly, it needs three pairwise relatively prime subkeys. To eliminate the requirement of entering these three subkeys as input, a method for creating three pairwise relatively prime subkeys from a single long key is proposed. The system key \( K \) can be of any length, but it is recommended to be at least 128 bits to resist brute force attacks. Firstly, the system key, \( K \), is divided into two parts called \( a \) and \( b \) with any reasonable lengths. For example, if the key is 128 bits, they can be of sizes 32 and 96 bits, respectively. Then, the subkeys are generated as follows:

\[
\begin{align*}
    k_1 &= ab + 1, \quad (6a) \\
    k_2 &= b k_1 + 1, \quad (6b) \\
    k_3 &= k_1 k_2 + 1. \quad (6c)
\end{align*}
\]

Given the Euclidean algorithm and the above construction method for \( k_2 \) and \( k_3 \), it is obvious that the three subkeys are pairwise relatively prime [7]. Furthermore, \( k_1 \) is constructed in a manner that ensures key sensitivity.

III. RESULTS AND DISCUSSION

This section discusses the security and the performance of the system and compares it to previous methods [8]–[12]. For the references that proposed more than one scheme, the compared schemes are the third scheme from [9], PM-CSR from [10], PM-MTD from [11], and the third scheme from [12]. The key used in the tests is \((8575062102FBCD4F357FBC5AF71A1BFC)_{16}\), the secret images used are shown in Fig. 2, and the secret shares generated are shown in Fig. 3.

Fig. 2: Secret images (a) \( I_1 \), (b) \( I_2 \), (c) \( I_3 \), and (d) \( I_4 \)

Fig. 3: Secret shares (a) \( S_1 \), (b) \( S_2 \), (c) \( S_3 \), and (d) \( S_4 \)

The histogram of an image shows the distribution of pixel values in the whole image. Figure 4 shows the non-uniform histograms of the secret images. Figure 5 shows the histograms of the three channels of the secret share \( S_1 \) separately, where the other secret shares have similar histograms. The histograms of the secret share show a uniform distribution of pixel values, which is the desired distribution for a strongly encrypted image [16].

\[
\text{Fig. 4: Histograms of secret images (a) } I_1, \ (b) \ I_2, \ (c) \ I_3, \text{ and (d) } I_4
\]

\[
\text{Fig. 5: Histograms of secret share } S_1 \ (\text{a) Red (b) Green, and (c) Blue}
\]

Entropy is the measure of randomness of image pixels, which implies unpredictability, and the value of entropy for an encrypted image is close to 8 [16] and calculated using the equation:

\[
\text{Entropy} = -\sum_{i=0}^{255} P(i) \log_2 P(i), \quad (7)
\]

where \( P(i) \) is the probability of the pixel value \( i \) in the image. Table I shows the entropy values for secret images and secret shares, where secret shares have entropies approaching the required value of 8.

<table>
<thead>
<tr>
<th>Secret Image</th>
<th>Entropy</th>
<th>Secret Share</th>
<th>Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_1 )</td>
<td>7.7624</td>
<td>( S_1 )</td>
<td>7.9998</td>
</tr>
<tr>
<td>( I_2 )</td>
<td>7.7622</td>
<td>( S_2 )</td>
<td>7.9998</td>
</tr>
<tr>
<td>( I_3 )</td>
<td>7.6698</td>
<td>( S_3 )</td>
<td>7.9998</td>
</tr>
<tr>
<td>( I_4 )</td>
<td>7.6385</td>
<td>( S_4 )</td>
<td>7.9998</td>
</tr>
</tbody>
</table>

Average 7.7082 Average 7.9998

The correlation coefficient, \( \rho \), measures the relationship between two vector variables, \( x \) and \( y \), of length \( n \). It has the range of \([-1, 1]\), where zero means no correlation and is the desired value for a good encryption system [16]. It is measured using:

\[
\begin{align*}
    \text{Cov}(x, y) &= \frac{1}{n} \sum_{i=1}^{n} (x_i - \frac{1}{n} \sum_{i=1}^{n} x_i)(y_i - \frac{1}{n} \sum_{i=1}^{n} y_i), \quad (8a) \\
    D(x) &= \frac{1}{n} \sum_{i=1}^{n} (x_i - \frac{1}{n} \sum_{i=1}^{n} x_i)^2, \quad (8b) \\
    \rho &= \frac{\text{Cov}(x, y)}{\sqrt{D(x)} \sqrt{D(y)}}. \quad (8c)
\end{align*}
\]
Table II shows the correlation between secret images and secret shares compared to previous schemes, where the results are good and comparable. Table III shows the correlation between secret shares compared to previous schemes, where the results are also good and comparable. It should be noted that some previous schemes only reported two or three Decimal Places (DP), which are not sufficient as the reported results are 0.00 and 0.000.

TABLE II: Correlation between secret images and secret shares compared to previous schemes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>I₁, S₁</td>
<td>0.0003</td>
<td>-0.0023</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
</tr>
<tr>
<td>I₂, S₂</td>
<td>0.0021</td>
<td>-0.0014</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>I₃, S₃</td>
<td>0.0006</td>
<td>-0.0017</td>
<td>-0.001</td>
<td>0.000</td>
<td>0.001</td>
<td>0.001</td>
</tr>
<tr>
<td>I₄, S₄</td>
<td>0.0004</td>
<td>-0.0004</td>
<td>-0.002</td>
<td>0.000</td>
<td>0.000</td>
<td>0.002</td>
</tr>
<tr>
<td>Average</td>
<td>0.0011</td>
<td>-0.0015</td>
<td>-0.001</td>
<td>0.000</td>
<td>0.000</td>
<td>0.001</td>
</tr>
</tbody>
</table>

TABLE III: Correlation between secret shares compared to previous schemes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>S₁, S₂</td>
<td>0.0435</td>
<td>0.0276</td>
<td>0.030</td>
<td>0.000</td>
<td>0.015</td>
<td>0.051</td>
</tr>
<tr>
<td>S₁, S₃</td>
<td>0.0589</td>
<td>0.0350</td>
<td>0.044</td>
<td>0.000</td>
<td>0.066</td>
<td>0.046</td>
</tr>
<tr>
<td>S₁, S₄</td>
<td>0.0343</td>
<td>0.0092</td>
<td>0.076</td>
<td>0.000</td>
<td>0.034</td>
<td>0.076</td>
</tr>
<tr>
<td>S₂, S₃</td>
<td>-0.0245</td>
<td>-0.0123</td>
<td>0.043</td>
<td>0.000</td>
<td>0.064</td>
<td>0.044</td>
</tr>
<tr>
<td>S₂, S₄</td>
<td>0.0494</td>
<td>0.1634</td>
<td>0.047</td>
<td>0.000</td>
<td>0.040</td>
<td>0.046</td>
</tr>
<tr>
<td>S₃, S₄</td>
<td>0.1156</td>
<td>0.0416</td>
<td>0.075</td>
<td>0.000</td>
<td>0.034</td>
<td>0.076</td>
</tr>
<tr>
<td>Average</td>
<td>0.0462</td>
<td>0.0441</td>
<td>0.056</td>
<td>0.000</td>
<td>0.040</td>
<td>0.056</td>
</tr>
</tbody>
</table>

Correlation coefficient is also measured between adjacent pixels in vertical, horizontal, and diagonal directions, where Table IV shows the correlation of the secret images and secret shares in all directions and channels and shows good results with low correlation. Also, scatter diagrams are drawn between adjacent pixels, where Fig. 6 shows the scatter diagrams of secret images in different directions and Fig. 7 shows the scatter diagram of secret share S₁ in different directions and channels, and other shares have similar scatter diagrams.

Root Mean Square Error (RMSE) measures the difference between two images and has the range of [0, 255] where larger values mean more difference between the images [8]. It is calculated using:

\[
RMSE = \sqrt{\frac{1}{W \times H} \sum_{i=1}^{W} \sum_{j=1}^{H} (x(i, j) - y(i, j))^2},
\]

where \(x(i, j)\) and \(y(i, j)\) are the pixels at \(i^{th}\) row and \(j^{th}\) column of the two images and \(W\) and \(H\) are the dimensions of the images. Table V shows the RMSE results between secret images and secret shares, which shows better results than the compared schemes. Table VI shows the RMSE results between secret shares, which shows comparable results with other schemes.

National Institute of Standards and Technology (NIST) SP-800-22 statistical test suite is a group of tests used to test the randomness of the produced images [17]. The P-value for each test must be greater than a significance level \(\alpha\), which may have a value in the range [0.001, 0.011], and the sequence length must be greater than \(10^6\) [17]. To perform this test, the
TABLE IV: Correlation of adjacent pixels

<table>
<thead>
<tr>
<th>Image</th>
<th>Vertical</th>
<th></th>
<th>Horizontal</th>
<th></th>
<th>Diagonal</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Red</td>
<td>Green</td>
<td>Blue</td>
<td>Red</td>
<td>Green</td>
<td>Blue</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>I₁</td>
<td>0.86607</td>
<td>0.76522</td>
<td>0.88095</td>
<td>0.92227</td>
<td>0.86386</td>
<td>0.90684</td>
</tr>
<tr>
<td>I₂</td>
<td>0.958409</td>
<td>0.96628</td>
<td>0.96940</td>
<td>0.93886</td>
<td>0.97152</td>
<td>0.97108</td>
</tr>
<tr>
<td>I₃</td>
<td>0.96633</td>
<td>0.98176</td>
<td>0.96641</td>
<td>0.96351</td>
<td>0.98110</td>
<td>0.96648</td>
</tr>
<tr>
<td>I₄</td>
<td>0.97332</td>
<td>0.96811</td>
<td>0.97359</td>
<td>0.94175</td>
<td>0.92905</td>
<td>0.94135</td>
</tr>
<tr>
<td>S₁</td>
<td>-0.00135</td>
<td>-0.00134</td>
<td>-0.00022</td>
<td>-0.00154</td>
<td>-0.00134</td>
<td>-0.00037</td>
</tr>
<tr>
<td>S₂</td>
<td>-0.00175</td>
<td>-0.00217</td>
<td>-0.00136</td>
<td>-0.00590</td>
<td>-0.00114</td>
<td>-0.00268</td>
</tr>
<tr>
<td>S₃</td>
<td>-0.00055</td>
<td>-0.00235</td>
<td>-0.00202</td>
<td>0.00024</td>
<td>0.00976</td>
<td>-0.00315</td>
</tr>
<tr>
<td>S₄</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

TABLE V: RMSE between secret images and secret shares compared to previous schemes

<table>
<thead>
<tr>
<th>Images</th>
<th>Proposed</th>
<th>[8]</th>
<th>[9]</th>
<th>[10]</th>
<th>[11]</th>
<th>[12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>I₁, S₁</td>
<td>92.82</td>
<td>10.77</td>
<td>92.81</td>
<td>92.79</td>
<td>92.85</td>
<td>92.72</td>
</tr>
<tr>
<td>I₂, S₂</td>
<td>100.44</td>
<td>10.58</td>
<td>100.04</td>
<td>100.01</td>
<td>100.04</td>
<td>100.04</td>
</tr>
<tr>
<td>I₃, S₃</td>
<td>100.58</td>
<td>9.95</td>
<td>100.40</td>
<td>100.34</td>
<td>100.28</td>
<td>100.37</td>
</tr>
<tr>
<td>I₄, S₄</td>
<td>92.31</td>
<td>9.45</td>
<td>92.36</td>
<td>92.27</td>
<td>92.21</td>
<td>92.40</td>
</tr>
<tr>
<td>Average</td>
<td>96.54</td>
<td>10.19</td>
<td>96.40</td>
<td>96.34</td>
<td>96.35</td>
<td>96.38</td>
</tr>
</tbody>
</table>

TABLE VI: RMSE between secret shares compared to previous schemes

<table>
<thead>
<tr>
<th>Images</th>
<th>Proposed</th>
<th>[8]</th>
<th>[9]</th>
<th>[10]</th>
<th>[11]</th>
<th>[12]</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₁, S₂</td>
<td>102.20</td>
<td>10.61</td>
<td>101.90</td>
<td>104.46</td>
<td>103.87</td>
<td>101.81</td>
</tr>
<tr>
<td>S₁, S₃</td>
<td>101.40</td>
<td>10.60</td>
<td>102.22</td>
<td>104.35</td>
<td>101.17</td>
<td>102.07</td>
</tr>
<tr>
<td>S₁, S₄</td>
<td>102.50</td>
<td>10.54</td>
<td>101.54</td>
<td>104.47</td>
<td>101.90</td>
<td>100.45</td>
</tr>
<tr>
<td>S₂, S₃</td>
<td>105.80</td>
<td>10.17</td>
<td>102.24</td>
<td>104.38</td>
<td>101.15</td>
<td>102.22</td>
</tr>
<tr>
<td>S₂, S₄</td>
<td>101.89</td>
<td>10.49</td>
<td>102.09</td>
<td>104.46</td>
<td>102.93</td>
<td>102.12</td>
</tr>
<tr>
<td>S₃, S₄</td>
<td>98.31</td>
<td>10.65</td>
<td>100.55</td>
<td>104.43</td>
<td>102.97</td>
<td>100.54</td>
</tr>
<tr>
<td>Average</td>
<td>102.05</td>
<td>10.60</td>
<td>101.59</td>
<td>104.42</td>
<td>102.50</td>
<td>101.53</td>
</tr>
</tbody>
</table>

same secret images were used but with size 1024 × 1024 and 24 bits for each random number, which is the pixel value of red, green, and blue concatenated. Table VII shows the test suite results for the secret share S₁, where all shares showed similar results of passing the fifteen NIST tests.

TABLE VII: NIST results for S₁

<table>
<thead>
<tr>
<th>Test</th>
<th>P-VALUE</th>
<th>Proportion</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Frequency</td>
<td>0.911413</td>
<td>0.958</td>
<td>✓</td>
</tr>
<tr>
<td>Block Frequency</td>
<td>0.996716</td>
<td>1.000</td>
<td>✓</td>
</tr>
<tr>
<td>Cumulative Sums</td>
<td>0.537191</td>
<td>1.000</td>
<td>✓</td>
</tr>
<tr>
<td>Runs</td>
<td>0.350485</td>
<td>1.000</td>
<td>✓</td>
</tr>
<tr>
<td>Longest Run</td>
<td>0.000879</td>
<td>1.000</td>
<td>✓</td>
</tr>
<tr>
<td>Rank</td>
<td>0.090936</td>
<td>1.000</td>
<td>✓</td>
</tr>
<tr>
<td>FFT</td>
<td>0.353476</td>
<td>1.000</td>
<td>✓</td>
</tr>
<tr>
<td>Non Overlapping Template</td>
<td>0.356572</td>
<td>0.988</td>
<td>✓</td>
</tr>
<tr>
<td>Overlapping Template</td>
<td>0.637119</td>
<td>1.000</td>
<td>✓</td>
</tr>
<tr>
<td>Universal</td>
<td>0.004301</td>
<td>0.917</td>
<td>✓</td>
</tr>
<tr>
<td>Approximate Entropy</td>
<td>0.285075</td>
<td>1.000</td>
<td>✓</td>
</tr>
<tr>
<td>Random Excursions</td>
<td>0.305346</td>
<td>0.972</td>
<td>✓</td>
</tr>
<tr>
<td>Serial</td>
<td>0.328019</td>
<td>0.958</td>
<td>✓</td>
</tr>
<tr>
<td>Linear Complexity</td>
<td>0.350485</td>
<td>0.958</td>
<td>✓</td>
</tr>
</tbody>
</table>

Differential attacks study how different inputs affect the output. Therefore, to test if a system is immune to differential attacks, the Least Significant Bit (LSB) of a random pixel of the input image is modified. Then, the Number of Pixel Change Rate (NPCR), the Unified Average Changing Intensity (UACI), and the Mean Absolute Error (MAE) are measured between the two images encrypted using the original input image and the slightly modified input image [18]. MAE is a measure of the absolute change and has the range of [0, 255], where higher values indicate high change. It is measured using:

\[ MAE = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left| x(i,j) - y(i,j) \right|. \] (10)

NPCR measures the percentage of different pixels and has the range [0, 100], where 100% means completely different images. It is calculated using:

\[ NPCR = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} D(i,j) \times 100. \] (11)

UACI measures the average intensity difference and has the range [0, 100], where 33.33% is the desired value [18]. It is calculated using:

\[ UACI = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \left| x(i,j) - y(i,j) \right| \times 100 \] (12)

Table VIII shows the average results of 50 test runs by changing a different random pixel in the secret image I₁, where the results are close to the values required for resisting differential attacks.

TABLE VIII: Measures for differential attacks

<table>
<thead>
<tr>
<th>Images</th>
<th>NPCR</th>
<th>MAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>S₁, M₁</td>
<td>99.61</td>
<td>85.32</td>
</tr>
<tr>
<td>S₂, M₂</td>
<td>99.61</td>
<td>85.32</td>
</tr>
<tr>
<td>S₃, M₃</td>
<td>99.61</td>
<td>85.34</td>
</tr>
<tr>
<td>S₄, M₄</td>
<td>99.61</td>
<td>85.33</td>
</tr>
</tbody>
</table>

It should be pointed out that NPC, UACI, and MAE were used by the previous schemes [8]–[12] as statistical measures and not as differential attack measures. This is because they measured them between secret images and secret shares, and between secret shares similar to what is given in Tables II, III, V, and VI.

As for the system key, it must be sensitive to one bit change, where such a small change should lead to a totally different retrieved image. In order to test key sensitivity, the LSB of the key is changed in the recovery process and the Mean Square Error (MSE) is measured between the original image and the wrongly retrieved image. Since the MSE is calculated using the square of (9), it has the range [0, 65025], where higher
values mean greater difference. Figure 8 shows the retrieved images and Table IX shows the results of MSE which are good.

**TABLE IX: MSE between retrieved images with correct key and retrieved images with modified key**

<table>
<thead>
<tr>
<th>Images</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1, M_{R_1}$</td>
<td>80.08.06</td>
</tr>
<tr>
<td>$R_2, M_{R_2}$</td>
<td>1011.57</td>
</tr>
<tr>
<td>$R_3, M_{R_3}$</td>
<td>1013.05</td>
</tr>
<tr>
<td>$R_4, M_{R_4}$</td>
<td>8518.14</td>
</tr>
<tr>
<td>Average</td>
<td>9343.11</td>
</tr>
</tbody>
</table>

Fig. 8: Retrieved images with modified key (a) $M_{R_1}$, (b) $R_2$, (c) $M_{R_3}$, and (d) $M_{R_4}$

In addition, the space and time complexity of the system depend on the dimensions of the images, where they are both of order $O(H \times W)$ because all operations are pixel wise and the number of secret images does not affect the system complexity. The runtime of the proposed system is measured on the setup (Windows 11 Home build version 21H2, Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz, 15.8 GB) using python programming language and JupyterLab IDE. Different number of inputs and different dimensions were used with 20 runs for each case, and the averages are reported in Table X which shows that the number of input images does not affect the time significantly.

**TABLE X: Sample runtimes for different image sizes**

<table>
<thead>
<tr>
<th>Inputs</th>
<th>Generation (sec.)</th>
<th>Recovery (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 256 x 256 images</td>
<td>0.29</td>
<td>0.29</td>
</tr>
<tr>
<td>4 256 x 256 images</td>
<td>0.29</td>
<td>0.29</td>
</tr>
<tr>
<td>2 512 x 512 images</td>
<td>1.12</td>
<td>1.12</td>
</tr>
<tr>
<td>4 512 x 512 images</td>
<td>1.14</td>
<td>1.14</td>
</tr>
<tr>
<td>2 1024 x 1024 images</td>
<td>4.49</td>
<td>4.46</td>
</tr>
<tr>
<td>4 1024 x 1024 images</td>
<td>4.57</td>
<td>4.69</td>
</tr>
</tbody>
</table>

**IV. Conclusions**

This paper presented a unified system for performing encryption and (n-n)-MSIS of colored images based on XOR, CRT, SHA-256 and S-box, where the same system was used for generation of secret shares and recovery of secret images. The three pairwise relatively prime subkeys used in the CRT were automatically generated from the sensitive system key. The secret shares passed all statistical evaluation criteria and gave good results for differential attack and key sensitivity measures. In the proposed system, the number of secret shares is the same as the number of secret images. In future research, the system can be improved to output any number of shares needed as the number of participants does not have to equal the number of secret images.

**ACKNOWLEDGMENT**

This paper is based upon work supported by Science, Technology, and Innovation Funding Authority (STIFFA) under grant number (#45631).

**REFERENCES**

Identification of Grape Leaf Diseases Using Proposed Enhanced VGG16

Gholamreza Farahani1, Seyedamirhossein Mousavi2, Ameneh Farahani3 and Hamidreza Farahani4
1 Department of Electrical Engineering and Information Technology, Iranian Research Organization for Science and Technology (IROST), Tehran, Iran
2 Department of Mechatronics, Azad Karaj University, Alborz, Iran
3 Department of Industrial Engineering, Ooj Institute of Higher Education, Qazvin, Iran
4 Department of Electrical Engineering, College of Technical and Engineering, Western Tehran Branch, Islamic Azad University, Tehran, Iran
farahani.gh@irost.org, amirmousavi228@gmail.com, st.a_farahani@riau.ac.ir, farahani.hamidreza@wtiau.ac.ir

Abstract—Due to the damage that various tree diseases cause to agricultural fields, there are always different methods to deal with them. This paper proposes a new algorithm to identify three common grapevine diseases: Downey Mildew, Anthracnose, and Powdery Mildew. This new algorithm is based on the Faster Region-based Convolutional Neural Networks (R-CNN) with an Enhanced VGG16 model. The proposed Enhanced VGG16 model can diagnose better these three types of diseases. Experimental results show that the proposed algorithm has 0.53%, 0.912%, 2.759%, and 7.268% improvement in the mean Average Precision (mAP) criterion compared to ResNet50, VGG16, GoogLeNet, and AlexNet networks, respectively. The precision of the proposed method is better than other methods, and the number of layers used in it is acceptable compared to other methods.

Keywords—faster R-CNN; VGG16; grapevine disease; mAP

I. INTRODUCTION

Diseases of garden trees can be a major problem for farmers. It can reduce the quality and quantity of the product, shorten the life of trees, and cause heavy economic damage to farmers and even the country's economy. Farmers can identify these diseases and eliminate them using a particular pesticide in the first stage. However, many human resources are required in extensive gardens. Therefore, it is time-consuming and costly. Moreover, constant exposure to these agricultural pesticides causes diseases in the person.

Among other problems, rainfall after each spraying stage can create the need for repair, and because of the high cost of pesticides, the garden requires great care and sensitivity. Therefore, by performing accurate and regular tracking operations, these harmful factors should be identified and implemented with a practical and managed program with the minimum consumption of pesticides.

There are diseases in all parts of agriculture of ornamental flowers [1], apple trees [2], walnut trees [3], and cherry trees [4], even in vineyards. These diseases appear on the leaves of tree trunks and even the fruit of these trees. But today, in the agricultural sector, grapes are an essential horticultural crop with a production of about 75 million tons per year [5]. It is one of the most common fruit crops globally, and Iran, with an annual harvest of 3 million and 167 thousand tons of grapes, has ninth place in producing this product in the world. Therefore, this product is of great importance in Iran [6]. The grapes garden is prone to diseases, and they are very destructive for grapes because they spread very quickly, and after the disease progresses, the treatments become ineffective. For the prevention of grapevine diseases, it is necessary to monitor them continuously. Nesteruk et al. applied machine learning for the plant classification of decompressed images. They divided a set of datasets into 18 classes, and the accuracy of 92.6% is promising in diagnosing plant diseases and deviation of plant phenology [7]. The paper is organized as follows: Section II describes related work. Section III expresses three famous grape leaves diseases. In Section IV, the data gathering and deep learning model are discussed. The proposed algorithm is explained in Section V. Experimental results for different methods are in Section VI. Finally, Section VII concludes the paper.

II. RELATED WORK


In the other research that Andrushia and Patricia carry out [11], at first, the captured images are pre-processed. Then noises and backgrounds of the images are removed, and features such as color and shapes are extracted. Then, feature selection, which is based on Artificial Bee Colony (ABC), is used to find the optimal feature set. Finally, these selected features are compared with other feature selection algorithms.

Gao et al. [12] used hyperspectral imaging for non-destructive detection of Grapevine Leaf Roll-Associated Virus-3 (GLRaV-3) during asymptomatic and symptomatic stages of GrapevineLeafroll Disease (GLD) in red-berried wine grape. This study used positive and negative Cabernet Sauvignon grapes for GLRaV-3. Also, Moore's dataset uses leaves of infected and non-infected vines separated in five phenological stages, and a hyperspectral imager acquired leaf images in the seasons of 2017, 2018, and 2019.
The results indicated that the hyperspectral imaging technique could non-destructively detect virus-infected grapevines during asymptomatic stages. In the system proposed by Adeel et al. [13], grapevine disease is identified with four steps. In the first step, a Local Contrast Haze Reduction (LCHR) enhancement technique is proposed to increase symptoms of local contrast. In the second step, LAB color transformation is held to select the best channel based on the pixel information that is later used in the thresholding function.

The Canonical Correlation Analysis (CCA) approach extracts color, texture, and fused geometric features and then fuses. At the time of feature fusion, a noise is added in the form of irrelevant and redundant features that are removed by Neighborhood Component Analysis (NCA). M-class SVM then performs the classification of final reduced features.

A. Convolutional Neural Network (CNN)

Paymode and Malode [14] proposed a method based on CNN for Multi-Crops Leaf Disease (MCLD) diagnosis. First, the features of the images are extracted. Then, diseased and healthy leaves with a method based on deep learning are classified using features. Finally, the VGG model based on CNN is used for performance improvement criteria.

Also, a system is presented for diagnosing papaya disease [15]. In this method, through a mobile application, a person takes a picture of the leaves suspected of papaya disease, sends the pictures to diagnose the disease, and compares the accuracy of the algorithms: Random Forest (RF), k-means clustering, SVC, and CNN. The system processes images and gives feedback. This intelligent system can detect papaya disease with high accuracy of about 98.4%.

Ahmad and Reddy [16] presented a system based on ML and smartphones to automate the disease detection process of plant leaves. The proposed system uses CNNs as deep learning to classify 38 categories of plant leaf diseases. They have used a dataset containing 96,206 images of healthy and infected plant leaves for training, testing, and validation. The second important convolutional neural network that can be mentioned is the AlexNet network, which was introduced by Alex Krizhevsky about fourteen years after LeNet, i.e., in 2012. This network has eight layers that are five convolutional layers and three fully connected layers [17].

Google introduced the inception structure at the 2014 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC14), being the best-performing model and is called GoogLeNet [18]. The GoogLeNet model can increase the network's performance by deepening the layers and achieving high accuracy. At the beginning of the Inception v1 module, where the input data is fed to three independent convolution layers, 1×1, 3×3, 5×5, and a 3×3 max-pooling layer, the outputs are combined into a single data set.

The VGG-Net [19] model reinforces that the CNNs must have a deep network of layers. ResNet [20], also known as Residual Net, shows significant performance in image classification and segmentation. The deep residual networks are like a bank of filters. ResNet is 8×deeper than VGG-Net with lower complexity. The ResNet with 152 layers won the ImageNet challenge 2015 (top-5 error of 3.57%).

III. GRAPE LEAVES DISEASES

There are various diseases in grape vineyards. For the grape tree, three Downey Mildew, Anthracnose, and Powdery Mildew diseases are commonly studied [21]. In continuation, these diseases are explained briefly.

A. Downey mildew

Downy mildew usually appears as spots on the top and bottom of the leaf and is related to fungal organisms that form in humid weather conditions. Usually, the color of this disease is gray to light purple. This disease needs high relative humidity to grow and cannot grow in dry weather [22].

B. Anthracnose

Grape anthracnose usually affects young leaves and stems, and then the leaves and stems affected by this disease distort and die. This disease appears as small brown spots and creates a pinhole effect. It Leads to wilting and loss of leaves [23].

C. Powdery mildew

Powdery mildew is a fungus that appears as a white layer on leaves and buds. This disease requires humid weather for reproduction, and it is seen mostly in older leaves. Lime Sulfur, ready-to-use Rose spray, and Rose spray concentrate are used to cope with this disease [24].

IV. DATA GATHERING AND DEEP LEARNING MODEL

A. Data Gathering

The images used for training and testing are 10,000 RGB photos that were taken manually from a vineyard near Shahriar city for two years (2018-2019) by a Kinect V2 sensor located 0.5 meters from the vine trees at heights of 30cm and 45cm. They are taken at different times of the day when the sunlight is changed. Their dimensions are 1920×1080 pixels.

All images are entirely taken manually, labeled, and stored in a separate file to improve the robustness of the proposed method. From these 10,000 images, 8000 are used for training and 2000 images for testing. The annotated images are augmented using geometric transformation and image enhancement.

All images were used as the inputs to Faster R-CNN-based Enhanced VGG16 (proposed method), AlexNet [17], GoogLeNet [18], VGG16 [19], and ResNet [20] model for training and testing. The test dataset includes an equal number of images (500 images) for each of the three diseases and 500 images for healthy leaves.

To accurately diagnose the type of grapevine disease, the proposed method identifies the exact type of disease if the symptoms of a particular disease are detected. Whereas increasing the number of training images, using data augmentation, one could prevent overfitting or non-convergence of the deep learning algorithm [25].
Data augmentation, including geometric transformation and data enhancement, was implemented using the software Matlab 2020a with the Image Processing Toolbox in the proposed method. The MATLAB function "imrotate" was used to rotate the raw image, and 90°, 180°, and 270° of rotation were achieved by changing the function parameter 'angle'.

For RGB to adjust the value of each component to increase or decrease the brightness of the image, a proportional factor close to 1 is multiplied by the original image [26]. It will be difficult for the operator during manual annotation because the edge of the target is unclear. Therefore, four proportional coefficients of 0.7, 0.8, 1.1, and 1.2 were selected based on the edge of the target, as shown in Fig. 1, which can be accurately identified during manual annotation.

After multiplying the mentioned proportional coefficients, if the resulting number is greater than 255, the same number 255 will be used. The histogram matching method has also been used to improve the quality of training sample images [27]. The camera may shake and have incorrect focus, which causes the image to be blurred; thus, it is difficult to detect. Therefore, the Matlab function 'imfilter' is used to blur the images with the generated filter in the training stage. The image blurry was used four times to make the convolutional network model strong adaptability to blurred images.

B. Deep Learning Model

The Faster R-CNN deep learning model [28] is widely used in agriculture, especially in identifying fruits, and it has good results. The Faster R-CNN model merges Region Proposals Network (RPN) object classification and localization into one unified deep object detection network. The RPN, a fully convolutional neural network, uses a partial convolutional layer of the VGG16 network to generate a feature map of a leaf's image and outputs a series of leaves target candidate regions [29].

An n×n sliding window is used to scan the feature map of the leaves image, and m target candidate regions are predicted for the position of each sliding window. The m proposals are used for the same localization that is called anchors. An anchor point is located in the center of the sliding window and is related to aspect ratio and scale [30].

In the proposed method, n=3, therefore m=9 anchors. Two fully connected layers of the same level regression and classification layers follow 512-dimensional features. The RGB image was used as the input of VGG16 [19] and then processed by the network. The network performs a random gradient descent method on the image blocks to update the parameters.

A filter of size 3×3 with stride one was used to construct a convolutional layer by VGG16, where the padding parameter in the same convolution is used as its parameter. Then a 2×2 filter with stride two was used to build the max-pooling layer. The feature map of the image was extracted through convolution, ReLU, and pooling operations, which were shared in the subsequent RPN layers and fully connected layers.

As shown in Fig. 2, the proposed disease grape leaf identification method is based on Faster R-CNN with an Enhanced VGG16 model. The dark blue, red, blue, green, and grey layers represent the full connection, convolutional, pooling, ReLU, and binary layers, respectively. The proposed Enhanced VGG16 model consists of 14 convolutional layers, five pooling layers, six Batch Normalization (BN) layers, and one fully connected layer. The following changes have been made to reach the Enhanced VGG from the VGG model. The Fe6 and Fe7 fully connected layers are removed from the VGG16 model. Conv6 and Global Average Pooling (GAP) layers are added to the VGG16 model. The two fully connected layers, Fe6 and Fe7, in the VGG16 fully connect each neuron with all the neurons in the previous layer. Therefore, they generate a significant number of parameters and occupy many computing resources. Reference [31] proposed a GAP for discarding these two fully connected layers. The GAP can be replaced with a fully connected layer. Experiments demonstrate that GAP can reduce the number of parameters, calculation volume, and the model's overfitting [31].

GAP calculates the average value of pixel points for each map. It outputs a feature point, fuses feature points into the feature vectors, and finally enters them into the Softmax layer. Therefore, the volume of operations, the number of parameters, and the overfitting are significantly reduced. Also, GAP can generate a feature graph for each category, and each feature graph and each category are connected more intuitively.

In this paper, to further optimize the VGG16 model, a convolutional layer Conv6 with a convolution kernel size of 1×1 has been placed in front of the GAP like many advanced network models such as GoogLeNet and ResNet that have introduced GAP. The effect of the number of filters on the model's performance during model construction has been analyzed by setting different 128, 256, and 512 filters. Conv5 of the VGG16 network model has changed to a deeply tandem group. In general, the VGG16 model continues the simple network structure of models like AlexNet. Nowadays, the use of deep tandem groups to improve network performance has been widely considered.
In this paper, using the inception structure of GoogLeNet [18], the three convolutional layers of the VGG16 network model Conv5 are transformed into a deeply tandem group.

As shown in Fig. 3, to increase the accuracy of the network, identification accuracy and for Conv5 to learn more characteristics; the input characteristics are transferred to all the convolution layers in this structure, and an improved structure is obtained. It improves the width of the network and increases the complexity of the model to a certain extent, and this created structure does not produce the waste step by step. The BN layers are added in the proposed Enhanced VGG16 [32].

The BN process is used for the input value in the neurons of the neural network to reach the standard normal distribution where the variance value is one, and the mean value is 0. This operation will put the input values in the sensitive areas of the non-linear function pairs. Therefore, very small changes in the input value can significantly affect the loss performance and can be used to eliminate the problem of gradient disappearance. Also, BN can significantly accelerate the model's convergence and training speed.

In 2015, the BN layer was added before the ReLU layer. However, some suggestions to add this layer after the ReLU layer are better. In [33], it has been shown that placing the BN layer in front of the ReLU layer sometimes causes damage to the model. However, increasing the number of training images using data enhancement can still avoid overfitting or underfitting. Avoid deep learning algorithm convergence. Placing the BN layer after the ReLU layer will positively improve accuracy and reduce losses. Therefore, in the method proposed in this paper, the BN layer is placed between the ReLU layer and the pooling layer to achieve the best results.

VI. EXPERIMENTAL RESULTS

In the implementation, the training platform included a computer with an Intel Xeon E7-8891 (3.50 GHz) ten-core CPU and a GPU of NVIDIA TITAN Xp Graphics Card 12GB (3584 CUDA cores) and 32 GB of memory, running on a Windows 10 64-bit system. The software tools included CUDA 11.4, cuDNN 8.0, Python 3.9, and Microsoft Visual Studio 16.0. The experiments were implemented in the Tensor Flow framework. The RPN was implemented as a fully convolutional network that was optimized through an end-to-end using backpropagation and mini-batch gradient descent.

The specific steps of training are as the following. A fixed value of 0.9 was set as the network's momentum, and 0.0005 was used as weight decay. Mini batches of 8000 images were used to train the model. The constant learning rate of 0.001 was chosen for all layers in the network. Iterations of 100,000 were selected to analyze the training process. In continuation, first, the criteria for evaluating the proposed method are expressed. Then the results of the proposed method with different recent methods are compared.

A. Criteria

Recall and Precision criteria are the primary and essential values for recognizing objects in the images. Precision refers to the ratio of correctly classified and placed categories to the total returned results. The Recall rate refers to the ratio of correctly classified and placed categories in the returned results to the total related categories.

Also, the criteria that are evaluated in this paper is average Precision (mAP) [34], which consists of Recall, Precision, and average. Precision and Recall are defined in (1) and (2), respectively.

\[
\text{Precision} = \frac{TP}{TP + FP} \quad (1)
\]

\[
\text{Recall} = \frac{TP}{TP + FN} \quad (2)
\]

where the number of correctly classified positive samples as a result of the prediction is shown as True Positive (TP), the number of samples that are misdiagnosed as a result of the prediction is shown as False Positive (FP), the number of undetected positive samples as a result of prediction shown as False Negative (FN).

According to the values obtained for the Precision and Recall criteria, the Precision-Recall curve can be accurately drawn with Precision as the ordinate and Recall as the abscissa. The area under the Precision-Recall curve is the Average Precision (AP). Therefore, the integral of the exact Recall curve is \(AP\) as (3).

\[
AP = \int_{0}^{1} p(r) \, dr \quad (3)
\]
The Recall criterion is divided into $n$ blocks $[0, 1/n, \ldots, (n-1)/n, 1]$, and then the $mAP$ is determined by the average value. If there are $N$ categories, the $mAP$ is calculated using (4).

$$ mAP = \left( \sum_{n=1}^{N} mAP(n) \right) / N $$  \hspace{1cm} (4)

### B. Compared methods

Multi-class grape leaf disease detection results for Enhanced VGG16, VGG16, GoogLeNet, ResNet50, and AlexNet methods are expressed in Table I. The proposed method has achieved the $mAP$ of 0.99625 for the three disease classes, the best value among other networks. The $mAP$ of the ResNet50, VGG16, GoogLeNet, and AlexNet are 0.99100, 0.98725, 0.96950, and 0.92875 respectively. The layers of the different networks are expressed in the last column of Table I.

Table II shows the percentage improvement of the $mAP$ for VGG16, GoogLeNet, ResNet50, and AlexNet networks compared to the Enhanced VGG16 (proposed method), which has the best $mAP$. Based on Table II, the proposed method has the $mAP$ improvement of 0.53%, 0.912%, 2.759%, and 7.268% than ResNet50, VGG16, GoogLeNet, and AlexNet networks, respectively. Therefore, ResNet50, VGG16, and GoogLeNet are in the second, third, and fourth ranks. The worst result is for AlexNet. From another point of view, the proposed Enhanced VGG16 model has improved the $mAP$ of the VGG16 model by 0.92%, which is an acceptable improvement. The Precision-Recall curves achieved by Enhanced VGG16 (proposed method), VGG16, ResNet50, AlexNet, and GoogLeNet on the testing dataset are shown in Fig. 4.

The Precision values of the proposed method are the highest, with the same Recall values for all infections and healthy leaves. The ResNet50 achieved the second-highest accuracy. VGG obtained a higher Precision for the three remaining networks than GoogLeNet and AlexNet on all three diseases and healthy leaves with the same Recall value. The lowest precision is for AlexNet. The advantages of the proposed Enhanced VGG16 model are it has the best $mAP$ and precision among other compared methods. However, the number of layers in the Enhanced VGG16 is not less than AlexNet and VGG16 models. Fig. 5 Shows the results of the loss curve of the test set for 100,000 iterations of the Enhanced VGG16 network (proposed method), which has the best results among other networks. The variation trend for Downey Mildew, Anthracnose, Powdery Mildew infections, and healthy leaves was almost overlapping. When the number of repetitions reaches 60,000, the average value of the loss becomes 0.512. After 60,000 repetitions, the amount of loss decreases with the increase in the number of repetitions, and the network generally becomes stable. The obtained stability confirms that the Enhanced VGG16 network has the characteristic to learn appropriately with good convergence ability.

### VII. CONCLUSION

The disease of vineyard leaves can spread quickly and cause a lot of damage to the vineyard. Therefore, identifying the type of diseases in the vineyard is vital to tackle its spread.
REFERENCES


Abstract Demonstrations and Adaptive Exploration for Efficient and Stable Multi-step Sparse Reward Reinforcement Learning

Xintong Yang, Ze Ji
School of Engineering
Cardiff University
Cardiff, UK
yangx66@cardiff.ac.uk, jiz1@cardiff.ac.uk

Jing Wu, Yu-Kun Lai
School of Computer Science and Informatics
Cardiff University
Cardiff, UK
wuj11@cardiff.ac.uk, laiy4@cardiff.ac.uk

Abstract—Although Deep Reinforcement Learning (DRL) has been popular in many disciplines including robotics, state-of-the-art DRL algorithms still struggle to learn long-horizon, multi-step and sparse reward tasks, such as stacking several blocks given only a task-completion reward signal. To improve learning efficiency for such tasks, this paper proposes a DRL exploration technique, termed A2, which integrates two components inspired by human experiences: Abstract demonstrations and Adaptive exploration. A2 starts by decomposing a complex task into subtasks, and then provides the correct orders of subtasks to learn. During training, the agent explores the environment adaptively, acting more deterministically for well-mastered subtasks and more stochastically for ill-learnt subtasks. Ablation and comparative experiments are conducted on several grid-world tasks and three robotic manipulation tasks. We demonstrate that A2 can aid popular DRL algorithms (DQN, DDPG, and SAC) to learn more efficiently and stably in these environments.

I. INTRODUCTION

Deep Reinforcement Learning (DRL) has achieved exciting advances recently in robotics [1]. However, DRL agents still struggle to solve many robotic tasks with long horizon, multiple steps and sparse rewards. For example, in Fig. 1, a robot is asked to push a block into a closed chest, where the chest needs to be opened before the block can be pushed into it, and only a reward is given when the task is done. Fortunately, humans provide many useful insights for mastering such tasks. In this paper, we revisit and integrate two ideas inspired from human experiences to make DRL more efficient and stable in these difficult tasks.

In the real world, humans benefit substantially from decomposing a complex task into a sequence of subtasks. This helps divide the task into a number of steps that are easier and faster to solve [2]–[4]. We can then accomplish the complex task by achieving each step in a specific order, such as following an instruction manual to assemble different parts of a piece of furniture. In addition, skills developed during learning one step can be reused in other steps, such as skills to stack a specific block can be reused to stack other blocks. The example in

Fig. 1 shows that the three steps of the pushing task are easier to achieve individually and they share similar arm motions (reaching and pushing).

Inspired by this, we propose firstly to leverage abstract demonstrations, which provide the correct order of steps to a learning agent instead of low-level motions. It has been shown that, given an efficient enough algorithm to learn the subtasks such as hindsight experience replay, abstract demonstrations can accelerate learning for multi-step tasks [5]. The main benefits of abstract demonstrations when compared to demonstrations of motion trajectories [6], [7] are: 1) they do not encode a specific pattern of the behaviours when solving a task; 2) they release robotic operators from the tedious processes of collecting motion trajectory data.

The second idea is inspired from humans’ adaptive exploration strategies. For example, children do not try to pick and stack Lego blocks randomly when they have already known which blocks they need and how to stack them together. This relates closely to the trade off between exploration and exploitation, a core issue in DRL [8]. Although exploration methods have been proposed for long-horizon tasks [9], its adaptivity has rarely been studied in the context of multi-step tasks. In multi-step tasks, the learning progress of later steps depends heavily on former steps. For example, stacking the fourth block requires the previous three blocks to have been stacked well. As a result, an agent that explores constantly or decays exploration in a task-agnostic way, which is the default setting in most works [10], will have difficulty approaching later steps. With this in mind, we propose to adjust exploration adaptively based on an agent’s performance on each of the task
steps.

Previously, these ideas have been explored individually (see a detailed review in Section II). In this paper, we highlight the effectiveness of their integration into DRL algorithms for learning multi-step, sparse reward tasks. We conducted a series of ablation studies on three representative off-policy DRL agents, i.e., DQN (deterministic, discrete actions) [11], DDPG (deterministic, continuous actions) [12] and SAC (stochastic, continuous actions) [13], using the Mini-grid [14] and Pybullet Multigoal environments [15].

The results show that abstract demonstrations can accelerate learning and improve performance slightly in the gridworld tasks (DQN) and significantly in continuous control tasks (DDPG and SAC). Adaptive exploration does not further improve task success rates in general, but it does stabilise learning with reduced performance variances. In short, our contributions include:

- We propose A², a method which integrates abstract demonstrations and adaptive exploration, for more efficient and stable multi-step sparse reward DRL.
- We discuss possible implementations of the adaptive exploration component of A² for modern DRL algorithms, including discrete and continuous actions, deterministic and stochastic policies.
- We validate the effectiveness of A² in various settings: with a deterministic DQN agent (discrete actions, gridworld tasks), a deterministic DDPG agent (continuous actions, robotic tasks) and a stochastic SAC agent (continuous actions, robotic tasks).

**Paper organisation:** Section II draws connections between A² and existing works. Section III briefly reviews important preliminaries for using A². Section IV describes A² in details. Section V presents experimental results and discussions. Section VI concludes the paper.

**II. RELATED WORK**

**Long-horizon multi-step tasks** have been addressed by task and motion planning (TAMP) [2]. TAMP typically generates high-level task sequences and plans motions for each subtask individually [3], [16]. Although the idea of task decomposition based on human priors has been around for decades [17], researchers have just started recently to study how it can help learning-based methods [18], [19]. Many works in RL focus on learning sub-policies (or options) that are specific to different subtasks or manifest different skills [20], [21]. Our work differs from them in that we decompose a task into subtasks to guide the learning of a control policy (as demonstrations), while they require an agent to learn to decompose a task during learning. In principle, our method can be incorporated with any task decomposition method, be it based on human priors or learning [22].

**Learning from demonstrations (LfD)** is a practical approach to teach robots behaviours using trajectories collected by another agent, typically a human. These behaviours are normally difficult to program [7]. DRL also benefits from demonstrations in sparse reward and/or long-horizon tasks (also known as imitation learning) [6]. Different from the mainstream works, which use trajectories at the control command level [9], we use an abstract form of demonstrations, i.e., the correct sequence in which task steps should be performed and learnt. This reduces the human labour spent on collecting long robot motion trajectories. For cases where some subtasks are too difficult to learn, motion trajectories can still be used to further accelerate learning for the subtasks. In other words, abstract demonstrations could be an alternative or a supplement to motion trajectories demonstrations.

**Automatic curriculum learning (ACL)** has become increasingly active in the field of RL and robotics [23]. Briefly, ACL methods automatically adapt the distribution of training data for a learning agent [23]. Our method can be seen as a contribution to ACL, which 1) prompts the agent to learn from easier subtasks to harder subtasks in the correct sequences (abstract demonstrations) and 2) adjusts the agent’s exploration behaviours based on its performance on each subtask.

**III. PRELIMINARIES**

**Markov Decision Process** is a tuple $(S, A, R, p, γ, ρ_0)$, where $S$ is the state space, $A$ the action space, $R(s, a)$ the reward function, $p(s' | s, a)$ the system transition model, $γ$ the discount factor and $ρ_0$ the initial state distribution. A policy $π(a | s)$ is a mapping from states to actions. A state-action (Q) value function, $Q^π(s_0, a_0)$, is defined as the expected, discounted and accumulated rewards starting from taking an action $a$ at state $s$ and following a policy $π$ thereafter, i.e., $Q^π(s_0, a_0) = E_{a_0, a_1, s_1, s_2, \ldots \parallel T} \gamma^T R(s_t, a_t)$, where $T$ is the maximal number of task timesteps. The goal of an RL algorithm is to find an optimal policy that maximises the value function [8].

**Goal-conditioned Reinforcement Learning:** The GRL problem operates on an MDP augmented with a goal space $G$ [24]. Instead of pursuing a single goal (conveyed by a single-objective reward function) as in standard RL, a GRL agent seeks to maximise a universal value function, $Q^G(s, g, a)$. Commonly, a goal is defined as some kind of transformation of a state, $g = m(s)$, assuming that, at any state, there is always a goal that is achieved if the agent arrives at that state.

**Deep Q Network:** DQN is an off-policy, deterministic RL algorithm that uses neural networks to approximate the Q function for discrete action tasks [11]. Commonly, DQN uses the epsilon-greedy (EGr) method for exploration. It takes a random action with a probability $ε$ and takes an action according to the learnt Q function with a probability $1 - ε$.  

**Deep Deterministic Policy Gradient:** DDPG is an off-policy actor-critic style DRL algorithm developed for continuous control tasks [12]. It approximates a Q-function and a deterministic policy using separate neural networks, alternating the updates of the two with some interval. Originally, Ornstein–Uhlenbeck noises are added to the learnt policy to form exploratory actions [12]. Recently, researchers tend to use a simpler, but efficient exploration strategy, which takes random actions with a probability of $ε$ and takes learnt actions with Gaussian noises with a probability of $1 - ε$ [24]. For
convenient reading, we name this strategy epsilon-Gaussian (EGa).

**Soft Actor Critic:** SAC is an off-policy, stochastic DRL algorithm [13]. It updates the policy and value network with entropy regularisation. Typically, the policy network outputs the mean and standard deviation for a Gaussian policy. SAC controls its exploration by adjusting the temperature parameter $\alpha$ either manually or automatically. One can also alter the standard deviation of the policy to directly control exploration in a heuristic manner.

**Hindsight Experience Replay:** HER is a goal-relabelling technique for GRL, which copies transitions and replaces their desired goals with some other goals obtained by a sampling strategy [24]. It improves learning significantly in goal-conditioned sparse reward tasks. Originally, Andrychowicz et al. proposed in [24] the final, random, episode and future strategies to sample goals for replacement, and demonstrated that the future strategy is the most efficient. It copies a transition $k$ times, samples $k$ future transitions, and replaces the desired goals for each copied transition with the sampled ones. We adopt the future strategy in all our experiments, with $k = 4$.

### IV. METHODS

This section illustrates our proposed $A^2$ method in details. Overall, it includes two components. First, abstract demonstrations are provided to guide the robot to learn subtasks in the correct order, leading to the completion of a task. Secondly, an adaptive exploration technique adjusts the exploration parameters to achieve faster and more stable learning.

#### A. Abstract demonstrations

We propose to first decompose a multi-step task into a sequence of subtasks that are easier to accomplish. We observed that, in many tasks (e.g., stacking blocks), although the decomposed steps may have different purposes (e.g., stacking each of the blocks), the underlying behaviours achieving them often share common characteristics (e.g., move-pick-move-place). This inspired us to use goal-conditioned reinforcement learning (GRL) [24] as our main learning framework. The reason is that GRL was proved to have the potential to learn shared representations and knowledge between the steps [18], and Hindsight Experience Replay (HER) enables a GRL agent to learn short-term goals efficiently with sparse reward signals [24].

The motivation of using abstract demonstrations over demonstrations at the control command level (kinematic demonstrations) is twofold. First, abstract demonstrations do not encode a specific pattern of behaviours for solving a subtask. Second, it releases humans from the tedious processes of collecting demonstrations of motion trajectories. However, in theory these two kinds of demonstrations could be used together in cases where the subtasks are difficult to learn without kinematic demonstrations. Given a task decomposition scheme, we then leverage human priors to label the correct sequence to achieve the subtasks. Regarding implementation, an abstract demonstration is represented by an ordered sequence of indexes, each corresponding to a subtask.

At the beginning of a learning episode, a final goal is sampled and given to the GRL agent. When abstract demonstrations are used, the agent is asked to learn the subtasks following the order given by the demonstration, instead of being directly given the final goal. The agent is given the next subtask if the current goal has been achieved. When given a new goal, the previous transitions within this episode are copied as a new trajectory and their desired goals are replaced with the new goal. New transitions associated with the new goals are then appended to the new trajectory, leaving the previous trajectory unchanged. This is to guarantee that the agent can finally learn to achieve all the goals without demonstrations. The episode ends when the number of maximum timesteps is reached.

We use a hyperparameter, $\eta \in [0, 1]$, to control the proportion of training episodes that are demonstrated. We follow the training procedure used in [24], which is organised in epochs, cycles and episodes. Each epoch has a number of cycles, $I$, and each cycle has a number of episodes, $J$. The number of demonstrated episodes in a cycle is computed by $\text{episode}_\text{demo} = \eta \times J$.

#### B. Adaptive exploration

As mentioned above, the idea of adaptive exploration in multi-step task learning is to reduce unnecessary exploration for well-learnt steps and increase on unfamiliar steps. Coupled with abstract demonstrations, this leads to an adaptive curriculum that guarantees the GRL agent to proceed learning towards the final goal as quickly as possible. Reducing unnecessary exploration also leads to more stable performance. Since deterministic and stochastic agents explore differently, we provide different implementations for them as follows.

**Deterministic** agents explore the environment using a base behavioural policy, $\pi_o(a|s, g)$, that contains some hyperparameters to control the exploratory behaviours. In this paper, we use the epsilon-Gaussian (EGa) strategy as our base behavioural policy for the DDPG agent, and the epsilon-greedy (EGr) strategy for the DQN agent.

The EGr strategy takes a random action with a probability $\epsilon$ or takes an action that maximises the learnt Q-function. In practice, $\epsilon$ will decay to a lower bound during the course of training, commonly using the following equation:

$$\epsilon = \epsilon_{\text{end}} + (\epsilon_{\text{start}} - \epsilon_{\text{end}}) \times e^{-\frac{n}{\beta}}$$

where, $\epsilon_{\text{start}}$ and $\epsilon_{\text{end}}$ are the upper and lower bounds, $n$ is the total elapsed environment timesteps and $\beta$ is the decay coefficient parameter. Intuitively, $\epsilon$ decays as the number of elapsed timesteps grows and its speed is controlled by $\beta$. To make it adaptive, we simply replace the exponential term by a performance metric. Specifically, we construct an $N$-dimensional vector to store the value of $\epsilon$ for each task step at the $m$-th epoch, $\epsilon_m$. We then use an $N$-dimensional vector to record the test success rates for all task steps at the $m$-th
epoch, denoted as $S_m$, initialised to 0. We update $\epsilon_m$ using the following equations:

$$
\epsilon_{m+1} = \epsilon_{end} + (\epsilon_{start} - \epsilon_{end}) \times (1 - S_m).
$$

For continuous action cases, the EGa strategy samples a random action uniformly with a probability $\epsilon$ or takes an action generated by a learnt policy with noises from a fixed Gaussian distribution. That is,

$$
\pi_b(a|s, g) = \begin{cases} 
\frac{1}{|A|}, & \delta \leq \epsilon \\
\mathcal{N}(\pi(a|s, g), \sigma), & \delta > \epsilon
\end{cases}
$$

where $\delta \sim \mathcal{U}(0, 1)$ and $\pi(a|s, g)$ is a learnt, deterministic, goal-conditioned policy. By varying the hyperparameters, $\epsilon$ and $\sigma$, based on the performance of each task step at different training epochs, we have an adaptive exploration strategy. Again, we use $N$-dimensional vectors to represent these two hyperparameters for the $N$ task steps at the $m$-th epoch, $\epsilon_m$ and $\sigma_m$. At the beginning of a training process, these two vectors are initialised to their initial values, $\epsilon_0$ and $\sigma_0$ ($\epsilon_0 = 0.2$ and $\sigma_0 = 0.05$ in all our experiments). We use another vector to record the test success rates and update $\epsilon_m$ and $\sigma_m$ by:

$$
\epsilon_{m+1} = \epsilon_0 \times (1 - S_m), \quad \sigma_{m+1} = \sigma_0 \times (1 - S_m).
$$

Stochastic agents explore the environment by sampling from a learnt stochastic policy. As an example, the SAC agent uses a Gaussian policy whose mean and deviation are produced by a neural network. We propose to use the success rate as a scaling factor for the deviation for different task steps. Similar to Eq. 2:

$$
\sigma_{m+1} = \sigma_0 \times (1 - S_m),
$$

where $\sigma_m$ is the deviations for the learnt stochastic policy for different task steps, and $\sigma_0$ can be pre-defined constants or learnt by a neural network.

Performance metrics. To obtain the test success rates, we perform $K$ testing episodes for each task step after each training epoch and calculate the average success rate for each step. This test is run with abstract demonstrations to reflect the performance of achieving a step starting from the previous one. However, computing Eqs. 1, 2 and 3 with the vanilla success rate would result in bumping changes of the exploration hyperparameters. As the success rate is calculated only from the $K$ testing episodes, it may not reflect the true performance of the current policy, because $K$ is usually not large for the sake of reducing computations. Thus, we use the polyak-average [25] of the test success rate vector, $S_m$ to calculate Eqs. 1, 2 and 3 instead of the vanilla success rates. The running average is calculated with a parameter $\tau_S \in [0, 1]$ as follows:

$$
S'_m = (1 - \tau_S) \times S_{m-1} + \tau_S \times S_m
$$

V. RESULTS

To investigate the effect of the proposed $A^2$ method, we conduct a series of simulation experiments using the Mini-Grid [14] and PMG environments [15]. All performances displayed were the success rates of achieving the final goal without demonstrations, averaged over five random seeds. Specifically, we experimented on six multi-step, sparse reward tasks, including GridDoorKey (3 sizes), ChestPush, ChestPickAndPlace and BlockStack (see Fig. 2 for a visualisation). Codes available at https://github.com/IanYangChina/A-2-paper-code.

A. Task and implementation details

GridDoorKey task: The state representation consists of the $x$ and $y$ coordinates of the agent, the key and the door, the heading direction of the agent, and two binary variables indicating whether the agent is carrying the key and whether the door is opened. The goal representation is the $x$ and $y$ coordinates of the target. The reward function gives a value of 0 when the goal is reached and $-1$ otherwise.

Manipulation tasks: We use the ChestPush and ChestPick tasks with one block and the BlockStack task with two blocks from the Pybullet Multigoal (PMG) environments [15]. The state representation and reward function remain the same as the original paper. The original goal representation only consists of the target coordinates of the blocks in the world frame. We add the gripper location and its finger width to the goal representation for specifying the task steps regarding grasping and placing.

At the beginning of each task, the agent is given a desired goal which, for example, specifies the desired location of a block. If abstract demonstrations are used in an episode, the agent will instead be given a subgoal according to the step at hand. The decomposition schemes for the tasks are as follows:

- Gridworld: three subgoals, including 1) reach the key, 2) reach the door and 3) reach the target location;
- ChestPush: three subgoals, including 1) open the chest, 2) reach the block and 3) push the block into the chest;
• ChestPick: four subgoals, including 1) open the chest, 2) grasp the block, 3) move to the top of the chest and 4) drop the block into the chest;
• BlockStack: four subgoals, including 1) grasp the base block, 2) move the base block to the target location, 3) grasp the second block and 4) stack the second block.

**Network architecture:** The DQN network has three MLPs of sizes 64, 128 and 64. The actor and critic networks for the DDPG and SAC have three MLPs of sizes 256. All layers use ReLU activation, except for the output. All actors use hyperbolic tangent to activate the final layer and all critics (include DQN) have no activation on the output.

### B. Ablations

This subsection examines the effect of different parameter values of the $A^2$ method, namely, the percentage of demonstrated episodes $\eta$ and the adaptive exploration update rate $\tau_S$. We perform ablations on the GridDoorKey25x25 tasks with the DQN agent, ChestPush task with the DDPG agent and BlockStack task with the SAC agent.

From Figs. 3a to 3c, we see that adding demonstrations improves convergence speeds and performances in general. Adding demonstrations to 50% or 75% of training episodes has the highest performance gains across all agents and tasks (green and red lines). Interestingly, providing demonstrations to all the training episodes tends to hurt the performances (purple lines). This could be due to a lack of negative learning signals when trained in a fully-demonstrated way, given that the value function normally requires negative examples to distinguish good transitions from bad ones for more accurate value prediction.

The ablations of the adaptive exploration update rate were conducted with a fixed percentage of demonstrated episodes ($\eta = 0.75$). From Figs. 3d to 3f, we see that, varying the value of $\tau_S$ tends to have no obvious effects. However, a small value may slow down learning as shown by the gridworld experiment (the orange line in Fig. 3d).

### C. General performance

This subsection examines the general improvements gained by the proposed $A^2$ method. According to the ablation studies (subsection V-B), we add abstract demonstrations into 75% of the training episodes and use 0.3 for the adaptive exploration parameter, $\tau_S$, for improvement evaluation.

As shown by Fig. 4, in all experiments, by adding abstract demonstrations in 75% of the training episodes, the agent learns faster with higher performances. The improvement is more obvious in the robotic tasks. Notice that as the robotic task becomes more difficult (from subfigures 4d to 4f), the gap of success rates becomes larger compared to the vanilla algorithm (blue lines). This demonstrates that abstract demonstrations can provide vast improvement on multi-step tasks, because it significantly eases the agent from the difficulty of exploration in long task horizon with sparse rewards and subtask dependencies.

Adaptive exploration provides less obvious improvements on top of abstract demonstrations in terms of success rates. However, it clearly stabilises the learning performance, as it shows a smaller variance. This is probably due to its effect on reducing unnecessary exploration such that the agent could act more decisively on well-mastered subtasks.

### VI. Conclusion

We introduced $A^2$ – abstract demonstration and adaptive exploration – to aid reinforcement learning algorithms in multi-
step, long-horizon and sparse reward tasks. We showed in section IV that $A^2$ can be integrated in value-based algorithms with discrete actions (e.g., DQN) and actor-critic algorithms with continuous actions with a deterministic (e.g., DDPG) or stochastic (e.g., SAC) policy. We evaluated $A^2$ in discrete gridworld and continuous robot manipulation environments. Results in section V showed that abstract demonstrations in general speed up learning with higher success rates, with a more significant gain in continuous robot manipulation tasks, and the adaptive exploration module helps the agent to learn more stably.

A limitation of our method is the requirement of a manually designed task decomposition scheme. This could be addressed by learning to decompose a long-horizon task into subtasks. More sophisticated task assumptions such as image observations may also be considered to evaluate the effects of our method.

REFERENCES


Abstract—Smart home systems with AI planning functionality have the potential to improve the lives of users. However, there is an emerging expectation that users should better understand and trust the decision-making of AI systems. In this paper, a smart home battery system is developed with a supplementary explanation module that allows non-expert users to intuitively visualise the planning process and to better understand its recommendations. The module relies on a notion of contrastive explanations, related to iterative planning, allowing users to ask contrastive questions based on state- and action-constraints that may or may not be satisfiable. The system is intended for an experimental study where participants interact with the planning system and complete a questionnaire, with the research objective being to evaluate the usefulness of the explanation module.

Index Terms—planning, explainable AI (XAI), contrastive explanations

I. INTRODUCTION

Artificial intelligence (AI) technology is developing at a rapid rate, yet as AI systems become more complex, it is increasingly difficult for stakeholders to understand their decision-making processes [1]–[7]. The field of explainable AI (XAI) seeks to address this issue by improving human-understanding of AI methods [8], [9].

The authors of [10] identify several challenges faced by XAI research. Firstly, there are many subfields of AI and each has a need for explanations, yet much of the current XAI research has focused on machine learning, with other subfields (e.g. AI planning) receiving less attention. Secondly, AI systems exhibit many stakeholders, including non-expert end-users, yet much of the current XAI research has focused on expert stakeholders (e.g. machine learning experts). Thirdly, different stakeholders interact with AI technology in different ways, and in the case of non-expert stakeholders evaluating XAI research depends on an underlying AI system that can sufficiently engage users and thus give rise to a need for explanations. In this paper, we address some aspects of the above challenges by developing a smart home battery system for non-experts that relies on AI planning and offers an explanation module.

Planning is an important subfield of AI. For example, planning can navigate scattered robots to reach a predetermined formation [11] and reduce global vehicle scheduling times [12]. However, fully understanding planning algorithms and their solutions can be extremely difficult for non-experts, if not experts. The field of explainable AI planning (XAIP) is the subfield of XAI as it relates to planning [13]. One example is the work of Eifler et al., who in [14] propose an approach to XAIP inspired by planning as an iterative process. This setting refers to a type of human-in-the-loop planning where features of the planning problem (e.g. goals, preferences) are only partially understood. The basic idea is that these aspects of the planning problem can be refined through an iterative process of altering the problem, (re)planning, and then observing the outputs.

The notion of contrastive explanation is well-established in philosophy and social science [15]. The observation is that when humans seek explanations they do not ask simply why P? but instead ask why P rather than Q? where Q is some contrastive event known as a foil. In AI planning, a planning problem describes a state-action transition system with an objective (e.g. goal states) such that a solution is an optimal plan that specifies applicable actions to execute in order to achieve the objective. If we specify constraints on valid states or actions then it is possible to limit the space of valid plans such that every solution must satisfy the constraint. In the human-in-the-loop setting for example, such constraints may express user preferences. A natural application of contrastive explanations to the setting of AI planning then is to allow users to ask of the system questions of the form why did the system recommend this plan rather than one that satisfies constraint C?. Suitable explanations may be that the constraint is unsatisfiable, that the constraint is satisfiable but leads to higher cost, or that the constraint can indeed be satisfied with equal cost.

A planning system was proposed in [10] for scheduling a smart home battery so as to optimise home electricity costs. In this paper we extend this work by developing an interactive planning system that offers constructive explanations. There are three main contributions of this paper. First, we design and implement a fully interactive planning system. Second, we propose a notion of contrastive explanations based on state- and action-constraints. Third, we design an explanation module that allows users to request contrastive explanations, having both visual and textual representations. The system includes two variants: in one variant (for the treatment group) the XAI module is included, and in the other variant (for the control group) the XAI module is excluded. The rest of the paper is organized as follows. In the Section 2, we introduce the main concepts used in smart home battery planning. In Section 3, we...
give a formal definition of contrastive explanations in planning and describe the architecture of the system. In Section 4, we provide details of the interactive interface and how it is used by the two different user groups. In Section 5 we conclude with a short discussion.

II. PRELIMINARIES

In this section we recall the main definitions from [10] used for the back-end of a smart home battery planner. A (deterministic) Markov decision process or MDP is a tuple $(S,A,T,C)$ where $S$ is a set of states, $A$ is a set of actions, $T : S \times A \to S$ is a transition function, and $C : S \times A \to \mathbb{R}$ is a cost function. A finite-horizon MDP extends the standard MDP definition by including a (decision) horizon $t_{\text{max}} \in \mathbb{N}$ with $D = \{1, \ldots, t_{\text{max}}\}$ the set of timesteps. A (non-stationary) policy is a function $\pi : S \times D \to A$ where the cumulative cost of $\pi$ in state $s \in S$ at timestep $t \in D$ is defined as:

$$V(s,t,\pi) = \begin{cases} C(s,a) + V(s',t+1,\pi) & \text{if } 1 \leq t \leq t_{\text{max}} \\ 0 & \text{otherwise} \end{cases}$$

(1)

such that $a = \pi(s,t)$ and $s' = T(s,a)$. A policy $\pi^*$ is optimal if it minimises $V(s,t,\pi^*)$ for all $s \in S$ and all $t \in D$.

**Definition 1.** A battery scheduling problem is a tuple $(\beta, s_1, \lambda, t_{\text{max}}, U, P_T, P_E)$ where:

- $\beta \in \mathbb{R}^{>0}$ is the (battery) capacity constant
- $s_1 \in [0, \beta]$ is the (current) battery level
- $\lambda \in [0, \beta]$ is the (dis)charge rate per timestep
- $t_{\text{max}} \in \mathbb{N}$ is the horizon with $D = \{1, 2, \ldots, t_{\text{max}}\}$
- $U : D \to \mathbb{R}$ the (electricity) consumption forecast
- $P_T : D \to \mathbb{R}$ the (electricity) import price forecast
- $P_E : D \to \mathbb{R}$ the (electricity) export price forecast

Intuitively, parameters $\beta$, $s_1$, and $\lambda$ describe characteristics of the battery, while $U$ describes a prediction of home energy consumption, and $P_T$ and $P_E$ describe predictions of import and export energy tariffs.

**Definition 2.** Let $(\beta, s_1, \lambda, t_{\text{max}}, U, P_T, P_E)$ be a battery scheduling problem. A battery scheduling model is an MDP $(S,A,T,C,t_{\text{max}},s_1)$ where:

- $S = [0, \beta]$ is the set of (battery level) states
- $A = \{-1, 0, 1\}$ is the set of (battery) actions with 1 the charge action, -1 the discharge action, and 0 the no-op action
- $T : S \times A \to S$ is the transition function defined for each $s \in S$ and each $a \in A$:

$$T(s,a) = \min\{\beta, \max\{0, s + a\lambda\}\}$$

(2)

- $C : S \times D \times A \to \mathbb{R}^{>0}$ is the cost function defined for each $s \in S$ and $t \in D$ as:

$$C(s,t,a) = u_a^+ P_T(t) + u_a^- P_E(t) - C^*(t)$$

(3)

$$C^*(t) = \min\left\{u_{\text{max}}^+ P_T(t) + u_{\text{max}}^- P_E(t), u_{\text{min}}^+ P_T(t) + u_{\text{min}}^- P_E(t)\right\}$$

(4)

where, given $s$ and $t$:

$$u_a = \begin{cases} U(t) + \min\{\lambda, \beta s\} & \text{if } a = 1 \\ U(t) - \min\{\lambda, \beta s\} & \text{if } a = -1 \\ U(t) & \text{if } a = 0 \end{cases}$$

(5)

$$u_{\text{max}} = U(t) + \lambda$$

(6)

$$u_{\text{min}} = U(t) - \lambda$$

(7)

such that $x^+ = \max\{0, x\}$ and $x^- = \min\{0, x\}$ for any $x \in \mathbb{R}$

- $t_{\text{max}} \in \mathbb{N}$ is the horizon with $D = \{1, 2, \ldots, t_{\text{max}}\}$
- $s_1 \in S$ is the initial state

Intuitively, the transition function $T$ describes simple battery dynamics where charge and discharge actions have the effect of increasing and decreasing the battery charge level, respectively. The battery’s state of charge (SoC) in state $s$ corresponds to the value $\frac{s}{\beta}$. The cost of a charge action at timestep $t$ is determined by the (import and export) prices at $t$ along with the amount of surplus energy consumption at $t$, while the cost of a discharge action is determined by the prices at $t$ along with the amount of surplus energy supply at $t$. Note that more realistic battery dynamics, such as asymmetric charge/discharge rates or the impacts of battery efficiency, can be modelled by simple revisions to Equation 2. Please refer to [10] for full details on Definition 1 and 2.

III. DESIGN OF SMART HOME BATTERY XAI SYSTEM

This section introduces the design of the Smart Home Battery XAI system. Our research question is to examine if visual information alone is sufficient or if visual information with contrastive explanations is more useful for non-experts to understand a planner’s output. In order to achieve this, we design two versions of a user interface, one for each user group (which will be discussed in detail in the next section).

A. State-constraint and action-constraint

Figure 1 shows the dynamic process of planning without constraints, which is implemented as a search algorithm traversing the state space induced by the MDP in order to obtain the lowest cost plan, as introduced in Section II.

![Fig. 1. Planning without constraints where the yellow node is the initial state, the green node is a goal state, and the red path is an optimal plan $\pi^*$](image-url)
plan $\pi$ satisfies state-constraint $f$ if, for any execution of $\pi$ from initial state $s_1 \in S$, it is guaranteed that $s_i \in f(t_i)$ for every timestep $t_i \in D$ with $s_i$ the state at $t_i$.

According to Definition 3, a state-constraint $f$ is a function mapping timesteps to subsets of states, with $f(t)$ being the set of acceptable states at timestep $t$. Intuitively, for a policy $\pi$ to satisfy $f$, then no execution of $\pi$ should encounter a state $s$ at timestep $t$ such that $s \notin f(t)$. An illustration of planning with state-constraints is shown in Figure 2.

**Definition 4.** Let $S$ be a set of states, $A$ be a set of actions and $D$ be a set of timesteps. An action-constraint is a function $g : D \rightarrow 2^A$ where $g(t) \subseteq A$ is the set of acceptable actions at timestep $t \in D$. A plan $\pi$ satisfies action-constraint $g$ if, for any execution of $\pi$ from initial state $s_1 \in S$, it is guaranteed that $\pi(s_i) \in g(t_i)$ for every $t_i \in D$ with $s_i$ the state at $t_i$.

According to Definition 4, an action-constraint $g$ is a function mapping timesteps to subsets of action, with $g(t)$ being the set of acceptable actions at timestep $t$. Intuitively, for a policy $\pi$ to satisfy $g$, then no execution of $\pi$ should encounter a state $s$ at timestep $t$ such that $\pi(s) \notin g(t)$. An illustration of planning with action-constraints is shown in Figure 3.

**Definition 5.** Let $\pi$ be an optimal plan, $f$ be a state-constraint, and $g$ be an action-constraint. A contrastive question is a tuple $Q = (\pi, f, g)$ such that $\pi$ does not satisfy $f$ or $g$, with the contrastive explanation for $Q$ defined as:

$$
\Psi(Q) = \begin{cases} 
(\pi', c') & \text{if } \pi' \text{ exists} \\
\bot & \text{otherwise}
\end{cases}
$$

where $\pi'$ is a plan satisfying both $f$ and $g$ while minimising cost $c' = V(s_1, 1, \pi')$.

Intuitively, a contrastive question asks why plan $\pi$ rather some other plan $\pi'$ satisfying $f$ and $g$? A contrastive explanation then answers either: (i) that such a plan exists and has equal cost to $\pi$, i.e. $c' = V(s_1, 1, \pi)$, (ii) that such a plan exists but has greater cost than $\pi$, i.e. $c' > V(s_1, 1, \pi)$, or (iii) that no such plan exists. By definition it cannot be that $\pi' = \pi$ or that $c' < V(s_1, 1, \pi)$.

**B. System Architecture**

![System Architecture](image)

The architecture of the smart home battery system is shown in Figure 4, which is based on the Model-View-Controller (MVC) design pattern. The visualisation pages belong to the **View**. The back-end is divided into two modules: the module interfacing with the front-end belongs to the **Controller** and the other modules for data processing belong to the **Model**. After the user selects a specific operation in the front-end, the instruction is passed to the **Controller**, where the instruction finds the corresponding interface to call the module in the **Model** to address the specific task, and then the result is returned to the view according to the previous route. **Module Plan** contains the code related to planning and **Module problem** refers to the code related to the MDP. The **Dataset** is stores information associated with the user, such as questionnaire responses, which will be used in the evaluation.

The interface for the smart home battery system is shown in Figure 5 and consists of two panels: the **Visual Information** panel (Visualisation panel) on the right, and the **Textual Information** modules on the left.

**C. Visual Information Interface Design**

The Visual Information panel consists of three plots which show the **Scheduled battery modes**, **Battery charge level** and **Electricity costs** from top to bottom. The **Scheduled battery modes** plot is a scatter chart showing which action (mode) the battery should perform at each timestep. For example,
at 02:10–02:15 on 4 April the battery mode should be “discharge” and at 13:30–13:35 on 6 April the battery should be “charge”. The Battery charge level plot is a line chart showing the charge level of the battery at each timestep. The Scheduled battery modes and Battery charge level plots are precisely correlated. For example, if the battery is discharged commencing at 03:55 on 3 April, the battery charge level decreases and if the battery is charged commencing at 18:45 on 3 April, the battery charge level increases. The Electricity costs plot shows the electricity costs with and without use of a battery at each timestep.

D. Textual Information Interface Design

Users can access the system through a web browser. The Textual Information is made up of two panels, the Summary panel and the Why not ... ? panel. The Summary Panel contains six data items as well as the Schedule Battery button; the Period indicates which period of time the system is currently displaying; the Consumption shows how much power was consumed during that period; the Average price indicates the average electricity price during that period; the Cost tells the user how much it has cost under standard conditions; and the Cost with Battery column tells the user how much would costs with use of the battery. For example, the Summary Panel in Figure 5 shows that the customer consumed 93.64 kWh of electricity from 1 April 2021 to 7 April 2021, costing £10.80, with an average price of 11.53 pence per kWh. If the battery was used, the customer would only have been charged £6.62.

The Why not ... ? panel shown in Figure 6 allows the user to ask contrastive questions as per Definition 5. The outcome of re-planning under constraints can be easily compared with the original outputs in the Summary panel, corresponding to the contrastive explanation. The panel consists of three rows of input, two buttons, and four items of data. As shown in Figures 5 and 6, the original optimal plan without constraints has a cost of £6.62, whereas the optimal plan with constraints has a cost of £6.71. The user is then told via textual information that a plan satisfying the user’s constraints would cost £0.09 more than the original plan.

IV. USER EXPERIENCE

We divide our users into two groups: one is the Control group with only visual and summary information, and another is the Treatment group with visual and summary information as well as contrastive explanations.

A. Interaction for Users in the Control Group

(a) Zoom in on data
(b) Zoom out on data

The Control group interface is shown in Figure 7. The figure shows that users in this group can see Visual Information and limited Textual Information. The interactive functions of the control group are described next. In the Summary panel, the system provides the user with the function to select a time period. If the Schedule Battery button is clicked, a modal box will appear allowing the user to select a new time period to...
be observed. The system will always display a default setting (e.g., 1–7 April 2022) to start with. In the Visualisation panel, at the bottom of this area there is a Datazoom slider, which can be dragged to zoom in (see Figure 8(a)) and out (see Figure 8(b)) on the data in the Visualization interface.

There is a synchronous relationship between the three plots in terms of zooming in and zooming out operations. When a user selects a specific point on any plot, the data for all three plots corresponding to the same timestep will be displayed in a tooltip. For instance, selecting the timestep commencing 16:25 on 3 April Figure 7 results in a tooltip displayed with relevant details such as battery charge level and electricity price.

**B. Interaction for Users in the Treatment Group**

The Treatment Group have access to the Why not …? panel in addition to the functions available to the Control Group. In Why not …? panel the three lines of inputs allow users to: (i) specify a range of timesteps; (ii) specify an acceptable action during that range; (iii) specify lower and upper bounds on acceptable states during that range. Together these inputs allow users to specify state- and action-constraints as part of a contrastive question. For the timestep range, users may input the bounds manually, or by selecting corresponding points on the Scheduled battery modes plot. If a user opts for the latter, the selected points will replace the previous inputs. With the necessary inputs provided, the user can then submit the question by selecting the Ask question button, which triggers the generation of a contrastive explanation. The red line/dots indicate the original plan while the green lines/dots indicate the plan of the contrastive explanation. If a user wishes to ask a different contrastive question, then they can select the reset button to clear the previous inputs. As shown in Figure 9, the contrastive explanation is represented visually in each chart along with the original plan. Thus, users can visually compare plans for differences in planned actions, battery charge levels, and costs at each timestep.

V. RELATED WORK

The evaluation of XAI, especially XAI for non-experts, is challenging because there are no obvious performance measures, such as model accuracy for machine learning or runtime for AI planning. Meaningful evaluation of XAI often requires at least some form of evaluation with human participants. For example, the authors of [16] explained a decision tree model with the help of an explanatory tool, then designed a questionnaire to invite participants to evaluate the model according to its characteristics. Similarly, the authors of [17], [18] first conducted a theoretical study of AI planning, after which the previous research results were developed into an explanatory system [19]. Both studies evaluated the explanations using Likert-scale questionnaires, with the former using a 5-point scale and the latter using a 7-point scale.

In the case of [16], the authors did not develop a complete system but instead presented explanations of a machine learning model to users in a static way, with no user interaction in the process. Research in [17]–[19] was more systematic in that they first conducted theoretical work and then developed the theoretical results plus explanations into a system for users to experience and evaluate. These studies relied on a control group so that comparisons could more easily highlight the usefulness of the explanations.

VI. CONCLUSION

In this paper, we proposed an XAI system for AI planning based on scheduling of a smart home battery. Our XAI interface design was guided by the research question: is visual information alone sufficient or is visual information...
with contrastive explanations more useful for non-experts. The paper discussed how users can better understand the system and build trust with it through contrastive explanations and iterative planning.

Table I shows the questionnaire we will use for a planned study based on our smart home battery system. Our next step is to invite participants and to randomly divide them into the Control and Treatment groups. Our analysis and evaluation are to address the research questions posed earlier and to provide some deeper understanding about how contrastive explanations can influence a user’s understanding of an AI planning system. We also want to discover the degree to which visual information alone can achieve basic user understanding of the underlying system.

REFERENCES


Abstract— Learning from demonstration (LfD) has been developed and proved to be a promising method for transferring skill knowledge from human to robot. It is desired to have a demonstration device that can effectively map demonstrations to the robot’s motion to compensate for the correspondence problem in LfD. Thus, we presented a novel design of a gripper-like exoskeleton for robotic grasping based on Learning from Demonstration. The exoskeleton collected the displacement of its grippers, position, and posture information in the demonstration. This paper first presented the mechatronic design of the exoskeleton and then described the experiment for data validation. We illustrated the preliminary functionality of the exoskeleton by reproducing the demonstration trajectory on the Franka Emika robot.

Keywords- Learning from demonstration; exoskeleton; robotic grasping

I. INTRODUCTION

Learning from demonstration (LfD) is a technology that allows robots to learn new skills from demonstrations performed by human operators [1]. LfD benefits from intuitiveness and rapid deployment without complex programming, leading to a thriving research field and contributing to various applications, e.g., manufacturing [2] and robotic surgery [3].

LfD also has a great potential for development in the industrial sector. For example, LfD has been applied in precise manipulation in assembly [4]. In traditional robotic assembly, the robot is pre-programmed to follow a well-defined trajectory with exact moments and force demands when contacting assembly parts. These requirements bring two limitations to the traditional robotics assembly task. One is that pre-programmed robot can only work in a stable environment and is less adaptable to unforeseen situations. Another is that traditional robotic assembly needs complex robotic programming, increasing the worker’s workload and production costs [4]. In contrast, learning from demonstration is a promising method to solve these limitations. It allows manipulation skills transferred from human demonstrators to robots to manage different scenarios, making the robot more compliant. Also, from the users’ perspective, LfD is an intuitive way to rapidly develop robotics tasks without knowing much knowledge of robotic programming.

According to the demonstration approach, there are three types of LfD: kinesthetic teaching, teleoperation, and passive observation [5]. The characteristics of these three approaches are shown in Table 1. The mapping between human demonstration and robot action is a common problem in LfD, although the three approaches have different mapping difficulties. To compensate for the mapping problem, a device for demonstration with a similar configuration and function to the robot can be helpful, as it is easier and more efficient to decode the mapping between the demonstration and the robot’s action while making the demonstration process more intuitive to the user.

<table>
<thead>
<tr>
<th>Demonstration</th>
<th>Ease of Demonstration</th>
<th>High DOF</th>
<th>Ease of Mapping</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kinesthetic</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Teleoperation</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Observation</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Several tools such as the joystick, data grove and exoskeleton were used in various LfD implementations. Rebelo et al. [6] developed a serial kinematics exoskeleton with a complex structure consisting of a series of joints and links in the arm section to mimic the configuration of the human arm. The relative positions and orientations between...
each joint and the end-effector of the exoskeleton are obtained by forward kinematics. Combined with the indexed Cartesian mapping, the correspondence issue between the operator and robot can be solved. The exoskeleton also integrated multiple functionalities for precise teleoperation. Pierce et al. [7] developed a wearable haptic exoskeleton enabling force and vibrotactile feedback. In their exoskeleton design, a decoder in a geared DC motor is used to measure the angle of rotation between the index finger and thumb, which is mapped to the robot gripper. Fang et al. [8] presented a data glove with IMU sensors at each finger joint to record hand posture information, which is then mapped to the individual joints of the robot and the end-effectors. However, Kuklieski et al. [9] investigated and compared the performance of a data glove and a joystick device. Their experiment shows that the joystick was easier for the assembly demonstration, while the misleading effect caused by the difference between the human hand analogy and robot structure is an issue of the data glove.

### TABLE 2. CATEGORIES OF THE DEMONSTRATION TOOL

<table>
<thead>
<tr>
<th>Category</th>
<th>Pros</th>
<th>Cons</th>
<th>Literature</th>
</tr>
</thead>
<tbody>
<tr>
<td>data glove</td>
<td>• hand motion capture</td>
<td>• misleading effect</td>
<td>[8]</td>
</tr>
<tr>
<td></td>
<td>• Compact and comfortable to wear</td>
<td></td>
<td></td>
</tr>
<tr>
<td>hand/arm exoskeleton</td>
<td>• High accuracy</td>
<td>• complex structure</td>
<td>[6-7]</td>
</tr>
<tr>
<td></td>
<td>• Multiple functionalities</td>
<td></td>
<td></td>
</tr>
<tr>
<td>joystick</td>
<td>• Easy to use for simple tasks</td>
<td>• lack of tactile and force</td>
<td>[9-10]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>information</td>
<td></td>
</tr>
</tbody>
</table>

From the above investigation, we can summarise the advantages and disadvantages of different tools used in the LiD, as shown in Table 2. Choosing a suitable tool can improve the quality of the demonstrations. Most demonstration tools are designed to adapt the configuration and functionality of the robot being used, thus, facilitating the mapping between demonstration and the robot action. It also makes the demonstration more intuitive. We, therefore, designed a new gripper-like exoskeleton (Fig. 1) with the following characteristics and usages:

1) The exoskeleton has a similar configuration as the robot’s gripper, making mapping between the demonstration tool and the robot’s gripper straightforward.

2) The grippers of the exoskeleton are interchangeable. In practice, robots often need to change grippers depending on what kind of object is being grasped, so for consistency, the grippers on our exoskeletons are also interchangeable.

3) The movement of the exoskeleton can be tracked, including the overall position, posture information and the displacement of the gripper which represents the grasping stage.

4) The process demonstration should be intuitive for the operator. The operator should be able to perform demonstrations using the exoskeleton without any training. It is desirable that the operator can intuitively associate the exoskeleton from its appearance to its usage.

The rest of this paper is organised as follows: Section II describes the system configuration of the exoskeleton, including mechanical design, sensors and data collection and processing, and the composition of the experimental system. Section III presents the details of the simulation and experiment as well as the result and analysis. Finally, Section IV makes a conclusion of the paper and presents future work.

## II. SYSTEM CONFIGURATION

### A. Mechanical design

Based on the design characteristics as described in the introduction, we designed a wearable exoskeleton (Fig. 2(a), (b)) that has the same configuration of parallel gripping as the gripper of the Franka Emika robot. The mechanical design of the exoskeleton can be divided into three parts: the parallel gripping mechanism, the interchangeable gripper, and the scalloped-shaped handle at the bottom.

In the parallel gripping mechanism, a guide rod parallel to the displacement sensor is mounted on the exoskeleton, as shown in Fig. 2(b), (c). A linear bearing on the guide rod connects the gripper with the displacement sensor, allowing the gripper to move jointly with the slider on the sensor in a parallel direction. Fig. 2(d) illustrates the usage of the gripper. The operator can move the left part of the gripper via the semi-circular finger ring on the gripper’s base while the right part is fixed. The sensor measures the displacement of the slider driven by the gripper. Therefore, the gripper displacement can be measured. Besides, the linear bearing effectively reduces the friction between the gripper base and the guide rod, resulting in a more responsive and smoother movement.

For interchangeable grippers, we designed a standardised gripper base with two screw holes to ease the replacement of different grippers. Currently, we designed two types of grippers (Fig. 2(a)), a hard gripper and a soft gripper based on the Fin-ray effect. The tips of the gripper are made of silicone with different hardness (i.e., more solid silicone for rigid gripper), while the base of the gripper is 3D printed with PLA materials.

A scalloped-shaped handle based on the shape of the human hand was designed. It adapts well to the shape of the human hand. The scalloped-shaped contact surface with the hand not only makes the exoskeleton less likely to slip but also allows the weight of the exoskeleton to be distributed evenly over the hand, thus enhancing comfort in use. Besides, several holes for buttons and switches were reserved on the handle to help further modification to the exoskeleton.

In addition, the ArUco marker cube is mounted on top of the exoskeleton to prevent occlusion caused by the
exoskeleton’s body. Thus, it allows a more robust marker detection.

Overall, the compact design of the structure reduces the weight and space occupation of the exoskeleton while ensuring its functionality. The weight of the exoskeleton is 0.63kg. Besides, rapid prototyping techniques such as 3D printing and silicone moulding were used to accelerate the design and fabrication of the exoskeleton. Most importantly, the exoskeleton with grippers allows the operator to perform grasping demonstration intuitively.

B. Data collection system

As shown in Fig. 3, the data collection system consists of the following components: the exoskeleton with sensors and ArUco marker cube; an Arduino board; a USB camera for tracking the ArUco marker; and a laptop with data collection software.

One of the issues in a system with multiple sensors is the inconsistency of sampling frequencies among different sensors. In our data collection system, the sampling frequency of the displacement sensor and IMU sensor is about 800Hz and 500Hz, respectively, while the recognition frequency of the ArUco marker is even lower. The inconsistency problem brings a challenge for the time synchronisation of the sensors and camera. Therefore, we used a reference clock with millisecond resolution for synchronisation, shared by the sensors and camera. The output data are therefore accompanied by a timestamp. In subsequent data processing, data from different sources can be aligned according to the timestamp. In addition, The Franka Emika Robot sends and receives data at 1000 Hz. The data sent to the robot need to be converted to the same frequency. Therefore, linear interpolation with time alignment was used to process the data into a standard format that works for the robot. In addition, a smoothing filter was applied to reduce the noise of the data.

We also developed a host software for data collection and initial data processing. As shown in the right part of Fig. 3, The host software enables the transfer of sensor data based on serial port communication, the ArUco marker detection, and data saving. It is worth mentioning that the software only handles the data collection and initial processing, whereas the subsequent processing of the data (i.e., data interpolation, alignment of the data to timestamps, and visualisation of the data) was done using python scripts because of its convenience.

Details of the sensors and ArUco marker detection are described as follows: The displacement sensor (Miran, KFM Micro Slider Type Displacement Sensor) has a measurable distance range of 0 to 75 mm with a 0.05~0.1mm resolution and a force resistance of less than 0.5 N. In addition, the sensor outputs an analogue value in the range of 0-1023 depending on its physical measurement. The analogue output was processed on an Arduino Uno board and then transferred to the laptop. Based on the conversion ratio of actual and analogue values, the actual displacement of the grippers can be derived.

In addition to the displacement sensor, the exoskeleton has an IMU sensor (Wit, WT64C IMU sensor) placed in the middle of the exoskeleton. The IMU sensor is a device to calculate and output various information (e.g., posture information, angular rate and acceleration) of the body. Since the posture information is important in our system, only the roll, pitch and yaw of the exoskeleton were decoded and recorded from the IMU sensor.

The position of the exoskeleton can be determined by detecting ArUco markers. The OpenCV API was used for the ArUco markers detection. To improve the accuracy of the marker detection, we designed an ArUco marker cube in which all faces except the bottom face are attached with a marker. Regardless of how the operator rotates the exoskeleton, at least one face marker can be identified. If
more than one marker is detected, the final position (i.e., the centre of the marker cube) is obtained by averaging their positions. In addition, the camera for ArUco marker detection is calibrated to eliminate lens distortion.

C. Trajectory learning and generalisation

In robot grasping tasks, the robot needs to move the object accurately to the target point, which involves robotic trajectory planning. A well-planned trajectory enables the robot to complete the grasping task more safely and efficiently. In the LfD-based grasping task, using a suitable model for trajectory learning is critical. A powerful tool to represent periodic and discrete trajectories is the dynamic movement primitives (DMPs) model. It is effective for robot skills learning and has been widely used in robotics tasks such as robotic assembly [11] and wiping tasks [12].

The DMP model proposed by Ijspeert et al., [13][14] is:

\[
\begin{align*}
\dot{v} &= K (g - x) - Dv + (g - x_0) f(s) \\
\dot{x} &= v 
\end{align*}
\]

(1)

where \(K, D > 0\) are stiffness and damping factors and \(\tau > 0\) is a timing parameter for adjusting the trajectory duration, \(x_0\) and \(g\) are the start and end point of the trajectory. \(f(s) = \theta^T \Psi(s)\) is a combination of the normalised Gaussian functions \(\psi_i\), where \(\theta = [w_1, w_2, ..., w_n]^T\), \(\Psi(s) = [\psi_1, \psi_2, ..., \psi_n]^T\), and \(w_i\) is the weight of \(\psi_i\) and the expression of the state variable \(\psi_i\) is

\[
\psi_i = \frac{\varphi_i(s)}{\sum_{i=1}^{n} \varphi_i(s)}, \quad \varphi_i(s) = \exp(-h_i(s - c_i)^2),
\]

(2)

where \(c_i\) and \(h_i > 0\) are the centres and widths of radial basis functions \(\varphi_i(s)\). The transformation function (or forcing term) \(f(s)\) depends on the phase variable \(s\), which is calculated by a canonical system

\[
\tau s = -\gamma s, \quad \gamma > 0
\]

(3)

The converging time is modified by the factor \(\gamma\) to make sure \(s \rightarrow 0\) at the end of the trajectory for erasing the influence of \(f(s)\). The \(\theta\) is estimated by the target value \(f_{\text{Tar}}(s)\) of \(f(s)\) that is calculated by the demonstrated trajectory \(x_d\) and velocity \(v_d\):

\[
f_{\text{Tar}}(s) = \left(\tau v_d - K (g - x_d) - Dv_d\right)/(g - x_0). \quad (4)
\]

In our experiment, we used the DMPs model to generate the demonstration trajectory from the position data of the exoskeleton collected from the grasping demonstration.

D. Experimental system

As shown in Fig. 4, the experimental system consists of the exoskeleton, the experimental platform with a USB camera, a laptop for data collection and processing, and a Franka Emika robot for trajectory reproduction. The experimenter can perform the grasping demonstration on the experimental platform. After data collection and processing, the standard data were sent to the robot to reproduce the trajectory.

III. SIMULATION AND EXPERIMENT

A. The experiment procedure

The experiment was conducted to test the data collection function of the exoskeleton, the effectiveness of the data processing, and to implement basic Learning from
demonstration to the robot for trajectory reproduction. At the current stage, we have experimented with robot trajectory reproduction based on position information and robotic grasping separately but have not yet combined the two experiments. Therefore, we present the trajectory replication experiment in this paper because of its Representativeness.

The experiment process is shown in the flowchart in Fig. 6. It consists of a demonstration phase and a robot trajectory reproduction phase. In the demonstration phase (Fig. 5 (a)), the experimenter used the exoskeleton to grasp a cube using its gripper and placed it at the goal point. The displacement of the gripper tips, position information, and posture information were recorded during the demonstration and processed into standard data for the robot. In the trajectory reproduction phase (Fig. 5 (b)), the robot reproduces the trajectory based on the processed data. Furthermore, the actual movement trajectory was recorded during the reproduction to compare with the demonstration trajectory.

B. Experiment result and analysis

Fig. 7 shows the trajectory of the demonstration and robot reproduction. The reproduced trajectory is generally close to the demonstration, meaning that the robot followed the demonstration trajectory as expected. Hence, the position data collected by the exoskeleton is valid. However, some perturbations that appear at the beginning and end of the movement need further investigation.

The gripper displacement and posture information of the exoskeleton are shown in Fig. 8. The displacement of the gripper over time looks like a U-shape. The distance dropped abruptly at the start, which represents the closing of the gripper, and then the displacement remained stable as the cube was gripped and moved towards the desired position. On reaching the goal point, the gripper was opened to release the cube, thus increasing the displacement. Hence, the data collected by the displacement sensor match the motion of the gripper.

For posture information, there is a relation between the roll angle of the exoskeleton and the grasping process. It can be seen from the similarity between the result of it and those of the displacement sensor, as shown in Fig 8.
However, the relation between pitch, yaw and grasping process needs further analysis.

IV. CONCLUSION AND FUTURE WORK

In this paper, we present the design of a new exoskeleton as a demonstration tool for learning from demonstration. It has a similar configuration to the robot gripper, thus, facilitating the mapping from demonstration to robot action. The displacement of the gripper tip, position, and posture information were collected during the demonstration. We tested and validated the data collection function of the exoskeleton through trajectory reproduction experiments. In the near future, we will introduce a novel tactile sensor to our exoskeleton. We will then conduct more rigorous experiments to verify the functionality of the exoskeleton and to improve its mechatronic design.

References


Figure 8. The gripper displacement and posture information of the exoskeleton.
Dual Quaternion Based Finite-Time Tracking Control for Mechatronic Systems with Actuation Allocation

Lichao Sun
School of Education, Communication & Society
King’s College London
London, United Kingdom
lichao.sun@kcl.ac.uk

Yanpei Huang
Department of Bioengineering
Imperial College London
London, United Kingdom
yanpei.huang@imperial.ac.uk

Ziwei Wang
Department of Bioengineering
Imperial College London
London, United Kingdom
ziwei.wang@ieee.org

Bo Xiao
Department of Computing
Imperial College London
London, United Kingdom
b.xiao@imperial.ac.uk

Eric M. Yeatman
Department of Electrical and Electronic Engineering
Imperial College London
London, United Kingdom
e.yeatman@imperial.ac.uk

Abstract—This paper investigates the tracking control performance regulation and actuation allocation of mechatronic systems subject to coupling motions. In particular, the kinematic and dynamic model is described by dual quaternion, which captures the coupling effect between translation and rotation movements. Considering external disturbances and system uncertainties, a non-singular fast terminal sliding controller is then developed to ensure finite-time tracking performance. In addition, the unwinding problem caused by the redundancy of dual quaternion is addressed with the help of a novel attitude error function. Furthermore, an improved simplex method is designed for distributing the developed control commands to proper actuators. Numerical simulations demonstrate the effectiveness with respect to disturbance suppression, fast tracking, high accuracy, and finite-time stability of the proposed method.

Index Terms—Dual quaternion, sliding mode control, finite-time stability, control allocation.

I. INTRODUCTION

High-Performance control of mechanical systems has attracted concerns from the fields of space [1], [2], surgery [3], [4], and human-robot interaction [5], [6]. However, proximate interaction tasks can be featured by coupling effect between translation and orientation motions. As a result, unified control design that addresses such coupling issue is still challenging. By developing a velocity-free nonlinear controller based on dual quaternion, the relative position and attitude of a rigid body were globally asymptotic stable [7]. The subsequent works based on dual quaternion become promising, but lack the consideration of control performance. For faster convergence performance, the finite-time controller was realized via terminal sliding mode [8], [9]. However, the above control implementation might suffer from singularity. Recently, increasing plants are equipped with redundant sets of actuators. With respect to over-actuated systems, control allocation aims to distribute the desired control command among the redundant actuators. To achieve the optimum with certain constraints, the fixed, single-gimbal, and double-gimbal thruster configurations were considered in [10]. In [11], a power-optimal reaction wheel motor torque distribution strategy was developed to minimize the instantaneous electrical power requirement.

In this paper, a non-singular fast terminal sliding mode (NFTSM) control strategy is developed to ensure finite-time convergence of state errors. In addition, the unwinding problem caused by the redundancy of dual quaternion is addressed through a novel attitude error function. Furthermore, optimization algorithm is adopted in control allocation to alleviate the physical restrictions on actuation characteristics and limitation of the maximal force based on null space. The main contributions of this paper are twofold: 1) The coupling phenomenon between translation and orientation motions is systematically addressed by dual quaternion while ensuring finite-time and unwinding-free convergence. 2) Control allocation is modelled as an optimal quadratic programming, which extends the application of traditional pseudo-inverse method.

II. PRELIMINARIES

A. Dual Quaternion

In order to tackle 6-degree-of-freedom (DoF) tracking issue, we introduce dual quaternion \( \hat{q} \) to describe the translational and rotational motion simultaneously [12]: \( \hat{q} := q + \epsilon q' \), where \( q \) and \( q' \) denote the real and dual part, respectively. \( \epsilon \) is the dual operator such that \( \epsilon^2 = 0, \epsilon \neq 0 \). Regarding \( \hat{a} = a + \epsilon a' \)
\( \hat{b} = b + \epsilon b' \), the following properties are presented for later derivation:

\begin{align}
\hat{a} \pm \hat{b} &:= a \pm b + \epsilon (a' \pm b'), \\
\hat{a} \times \hat{b} &:= a \times b + \epsilon (a \times b' + b \times a'), \\
\hat{a} \circ \hat{b} &:= ab + \epsilon a'b', \\
\hat{a}^* &:= a + \epsilon (a' \circ b'),
\end{align}

(1)

\begin{align}
\hat{\omega} \times \hat{M} &:= \hat{c} \times \hat{M} - \hat{K}_{\text{sgn}}(\hat{S}) - \hat{M}_0 \hat{\Theta}(\hat{q}_e)^{-1} \left[ \hat{\alpha}_1 \circ \dot{\hat{\xi}}_e + \dot{\hat{\Theta}}(\hat{q}_e) \hat{\omega} e + \hat{\alpha}_2 \circ W(\hat{\xi}_e) \circ \hat{\Theta}(\hat{q}_e) \hat{\omega} e + \hat{\alpha}_3 \circ \text{sig}(\hat{S}) \cdot p_2 \right],
\end{align}

(17)

Considering the external disturbances and system uncertainties, kinematics and dynamics are equivalent to:

\( \hat{\xi}_e = \hat{\Theta}(\hat{q}_e) \hat{\omega}_e, \)
\( \hat{M}_0 \hat{\omega}_e = \hat{u}_c + \hat{M}_0 [\hat{\omega}_e \times (\hat{q}_e^* \circ \hat{\omega}_d \circ \hat{q}_e) - \hat{q}_e^* \circ \hat{\omega}_d \circ \hat{q}_e] \)
\( - \hat{\omega} \times \hat{M}_0 \hat{\omega} + \hat{\Phi}, \)

where \( \hat{\Theta}(\hat{q}_e) = \frac{1}{2}(\hat{q}_e I + \hat{\xi}_e^*) \) with \( \hat{\xi}_e = -0.5 \eta_0 \hat{\omega}_e \) and \( \cdot (\cdot)^* \) is the cross product matrix operator. \( \hat{M}_0 \) and \( \hat{\Phi} \) denote the nominal and uncertain part of \( \hat{M} \) such that \( \hat{M} = \hat{M}_0 + \Delta \hat{M} \) and therefore \( \hat{\Phi} \) can be written as:

\( \hat{\Phi} = -\Delta \hat{M} \hat{\omega} + \Delta \hat{M} [\hat{\omega}_e \times (\hat{q}_e^* \circ \hat{\omega}_d \circ \hat{q}_e) - \hat{q}_e^* \circ \hat{\omega}_d \circ \hat{q}_e] - \hat{\omega} \times \hat{\Delta} \hat{M} \hat{\omega} + \hat{u}_d. \)

(11)

To realize the control objective, the terminal sliding mode (TSM) [14] is modified as dual form:

\( \hat{S}_i = \hat{S}_i + \epsilon \hat{S}_i' = \hat{\xi}_e + \hat{b}_i \circ \text{sig}(\hat{\xi}_e)^* + \hat{b}_i \circ \text{sig}(\hat{\xi}_e)^* \)
\( \text{sign}(\cdot) \) is the sign function. \( \alpha \in (0,1) \)
\( \hat{b}_i = b + \epsilon b' \) with \( b_i \) and \( b'_i \) being positive constants. Based on terminal sliding surface (12), the conventional TSM controller can be then designed as the similar structure in [15]. Unfortunately, the above algorithm implementation might not be well posed since the negative index term containing may cause singularity. In consideration of physical input saturation of the practical actuators, the aforementioned controller outputs tend to result in instability of the closed-loop system.

Inspired by [16], a variant of the non-singular fast terminal sliding mode (NFTSM) in dual form is proposed to eliminate the singularity phenomenon and improve the convergence rate:

\( \hat{S}_i = \hat{\xi}_e + \hat{\alpha}_{1i} \circ \hat{\xi}_e + \hat{\alpha}_{2i} \circ \hat{\hat{S}}_i, i = 1,2,3 \)

(13)

where \( \hat{\alpha}_{1i} = \alpha_{1i} + \epsilon \alpha_{1i}, \hat{\alpha}_{2i} = \alpha_{2i} + \epsilon \alpha_{2i}, \alpha_{1i}, \alpha_{1i}', \alpha_{2i}, \alpha_{2i}' \) are positive constants. The auxiliary terminal sliding surface is designed as:

\( S'_{ai} = \hat{\xi}_e + \hat{\alpha}_{1i} \circ \hat{\xi}_e + \hat{\alpha}_{2i} \circ \text{sig}(\hat{\xi}_e)^* \)

where \( \hat{\alpha}_{1i} = (2 - p_1) \delta^{p_1 - 1}, \hat{\alpha}_{2i} = (2 - p_1) \delta^{p_1 - 1}, p_1 = (p_1 - 1) \delta^{p_1 - 2}, \delta \) and \( \delta' \) are small positive constants, \( p_1 \in (0.5, 1) \) is a positive constant. To achieve the control objective, an NFTSM based control law is proposed as:

\( \hat{u}_c = -\hat{M}_0 [\hat{\omega}_e \times (\hat{q}_e^* \circ \hat{\omega}_d \circ \hat{q}_e) - \hat{q}_e^* \circ \hat{\omega}_d \circ \hat{q}_e] + \hat{\omega} \times \hat{M}_0 \hat{\omega} - K \text{sgn}(\hat{\hat{S}}) - (\hat{M}_0 \hat{\Theta}(\hat{q}_e)^{-1}) \hat{\alpha}_{1i} \circ \hat{\xi}_e + \hat{\Theta}(\hat{q}_e) \hat{\omega}_e + \hat{\alpha}_{2i} \circ W(\hat{\xi}_e) \circ \hat{\Theta}(\hat{q}_e) \hat{\omega}_e + \hat{\alpha}_{3i} \circ \text{sig}(\hat{S}) \hat{\omega}_e \)

(17)
where $\dot{\alpha}_3 = \alpha_3 + \epsilon \alpha'_3$ and $\dot{K} = K + \epsilon K'$. $\alpha_3$, $\alpha'_3$, $K$, $K'$, and $p_2$ are positive constants. Diagonal matrix $W(\xi_e)$ is the auxiliary switching term, whose $i$-th entry is designed as:

$$W(\xi_{ei}) = \begin{cases} p_1 |\xi_{ei}|^{p_1-1} & \text{if } \dot{S}_i = 0 \text{ or } \dot{S}_i \neq 0, |\xi_{ei}| \geq \delta \\ r_1 + 2r_2 |\xi_{ei}| & \text{if } \dot{S}_i \neq 0, |\xi_{ei}| < \delta \end{cases}$$

and $p_2$ are positive constants. Diagonal matrix $W(\xi_{ei})$ is the auxiliary switching term, whose $i$-th entry is designed as:

$$W(\xi_{ei}) = \begin{cases} p_1 |\xi_{ei}|^{p_1-1} & \text{if } \dot{S}_i = 0 \text{ or } \dot{S}_i \neq 0, |\xi_{ei}| \geq \delta' \\ r'_1 + 2r'_2 |\xi_{ei}| & \text{if } \dot{S}_i \neq 0, |\xi_{ei}| < \delta' \end{cases}$$

Theorem 1: Consider dual quaternion based system (8) with external disturbances and system uncertainties that satisfy the bounded assumptions, within control law (17), then the relative dual quaternion and relative velocity motor are guaranteed to converge in finite time.

Proof. Consider the following Lyapunov function candidate: $V_1 = \frac{1}{2} |\dot{S}| M_0 \dot{S}$. Taking the time-derivative of $V_1$ and substituting (10) into it yield:

$$\dot{V}_1 = [\dot{S}] M_0 \dot{e}_c + M_0 \alpha_3 \otimes \dot{e}_c + M_0 \alpha_2 \otimes \dot{S}_a$$

(18)

By combining the controller (17) with (20) is given by

$$\dot{V}_1 \leq [\dot{S}] M_0 \alpha_2 \otimes \dot{S}_a - M_0 \Theta(\hat{q}_e) \dot{\omega}_e + M_0 \alpha_1 \otimes \dot{\omega}_e + M_0 \alpha_2 \otimes \dot{S}_a$$

(19)

Consider the following Lyapunov function: $V_2 = \frac{1}{2} |\dot{\xi}_{ei}| \dot{\xi}_{ei}$.

(20)

Then the derivation of $V_2$ with respect to time follows:

$$\dot{V}_2 = - \sum_{i=1}^{3} \left( \alpha_{1i} |\xi_{ei}|^2 + \alpha'_{1i} |\xi_{ei}|^2 + \alpha_{2i} |\xi_{ei}|^{p_1+1} + \alpha'_{2i} |\xi_{ei}|^{p_1+1} \right) \leq -\mu_2 V_2^{\frac{p_1+1}{2}}$$

(21)

Accordingly, based on Lemma 1, state variables are guaranteed to converge in finite time, namely $\xi_{ei} \rightarrow [0, 0, 0]^T + \epsilon [0, 0, 0]^T$ and $\dot{\omega}_e \rightarrow 0$.

Remark 1: The fast terminal sliding mode will switch into asymptotic one when state variables approach zero in (17). It is guaranteed that the state variables will converge within finite settling time during the sliding phase if $\mu_2$ and $p_1$ are chosen properly. It is further noted that $K$ and $K'$ are control gains, which are suggested to be large enough for the robust stability.

Remark 2: The traditional TSM controller tends to cause singularity because of the negative index term with the state variables converging to zero. Whereas, the proposed control law is non-singular due to the following facts:

1) $\dot{S} \neq 0$. State variables are not small enough to cause singularity. Thus, the control law (17) is obviously non-singular.

2) $\dot{S} = 0$. Then the control law can be re-written as

$$\ddot{u}_c = - \hat{M}_0 \hat{\omega}_e \times (\hat{q}_e^* \hat{\omega}_d \hat{q}_e) - \hat{q}_e^* \hat{\omega}_d \hat{q}_e + \hat{\omega} \times \hat{M}_0 \hat{\omega}$$

(22)

Therefore, the singularity will not occur in the case of $p_1 \in (0.5, 1)$.

Remark 3: Double value of quaternions makes the attitude unwinding, which descends the effectiveness and global stability of proposed controller. To eliminate the unwinding problem, two anti-unwinding attitude error functions were presented in [17]. However, discontinuity points exist if $\psi_1 = 2(1 - |q_0|)$ is adopted, and the respond rate declines depends on the attitude error if $\psi_1 = 2(1 - |q_0|)$.

IV. ACTUATION ALLOCATION

To improve reliability and safety, redundant actuators are often equipped with modern mechanical systems to provide corresponding forces and torques. Inspired by [18], consider the following constraint condition in dual framework

$$\ddot{u}_c = \hat{D} \otimes \hat{u}_a$$

(23)
where \( \hat{u}_a \) denotes the control vector applied in redundant actuators, and \( \hat{D} \) is control allocation matrix. Note that if there is no uncertainty due to the actuator faults, the pseudo inverse (PI) control allocation can be realized as follows

\[
\hat{u}_a = \hat{D}^T \odot (\hat{D} \odot \hat{D}^T)^{-1} \odot \hat{u}_c.
\]

A. Null Space Based Pseudo Inverse (NSPI) Control Allocation

The linear mapping between \( \hat{u}_a \) and \( \hat{u}_c \) is presented through the PI control allocation. However, the solution given by (28) may not satisfy the practical thruster range with the limitation of the thruster configuration [19]. Thus, the optimal solution can be improved by employing the null space of the control allocation matrix

\[
\hat{u}_a = \hat{D}^T \odot (\hat{D} \odot \hat{D}^T)^{-1} \odot \hat{u}_c + \hat{\zeta}
\]

where \( \hat{D} \odot \hat{\zeta} = \hat{0} \), namely Null(\( \hat{D} \)) = \{ \hat{\zeta} | \hat{D} \odot \hat{\zeta} = \hat{0} \}. Thus, the thruster output can be adjusted to the available range with the proper choice of \( \hat{\zeta} \). Furthermore, \( \hat{\zeta} \) can be expressed as: \( \hat{\zeta} = \hat{x} \odot \Gamma \), where \( \hat{x} = [\hat{x}_1, \hat{x}_2, ..., \hat{x}_{n-6}] \) is the basic solution of null space, and \( \Gamma = [\Gamma_1, \Gamma_2, ..., \Gamma_{n-6}]^T \) is the undetermined coefficient. Thus, the control allocation can be described by an optimization problem

\[
\min \ J = \sum_{i=1}^{n} < \hat{\zeta}_i | \hat{\zeta}_i > = < \hat{\zeta}_1 | \hat{\Gamma} >
\]

s.t. \( \hat{G}_1 \odot \hat{\kappa}_1 \leq \hat{Q}_1 \)

where \( \hat{\zeta}_1 = [\hat{x}_1 \odot \hat{x}_1, \hat{x}_2 \odot \hat{x}_2, ..., \hat{x}_{n-6} \odot \hat{x}_{n-6}] \), \( \hat{G}_1 = [\hat{x}, -\hat{x}] \), \( \hat{\kappa}_1 = [\hat{\Gamma}, \hat{\Gamma}] \), \( \hat{Q}_1 = [\hat{u}_{am} - \hat{D}^T \odot (\hat{D} \odot \hat{D}^T)^{-1} \odot \hat{u}_c, \hat{D}^T \odot (\hat{D} \odot \hat{D}^T)^{-1} \odot \hat{u}_c] \), \( \hat{u}_{am} \) is the maximum output capability of the actuators.

B. Simplex Method Based Improved Pseudo Inverse Control Allocation

The control allocation problem (30) can be rewritten as:

\[
\min \ J = \sum_{i=1}^{n} < \hat{\zeta}_i | \hat{\zeta}_i > = < \hat{\zeta}_2 | \hat{\kappa}_2 >
\]

s.t. \( \hat{G}_2 \odot \hat{\kappa}_2 = \hat{Q}_2 \)

where \( \hat{G}_2 = [\hat{G}_1, \hat{I}] \), \( \hat{\zeta}_2 = [\hat{\zeta}_1, \hat{0}] \), \( \hat{\kappa}_2 = [\hat{\kappa}_1, \hat{\kappa}_s]^T \), \( \hat{Q}_2 = [\hat{Q}_1, \hat{0}] \), and \( \hat{\kappa}_s \) is the slack variable.

**Theorem 2:** With regard to non-singular dual sub-square matrix \( \hat{G}_{2B} \), the programming problem (31) is equivalent to:

\[
\min \ J = \hat{c}_{2B} \odot \hat{G}_{2B}^{-1} \odot \hat{Q}_2
\]

\[
- (\hat{c}_{2B} \odot \hat{G}_{2B}^{-1} \odot \hat{G}_{2N} - \hat{c}_{2N}) \odot \hat{\kappa}_{2N}
\]

s.t. \( \hat{\kappa}_{2B} + \hat{G}_{2B}^{-1} \odot \hat{G}_{2N} \odot \hat{\kappa}_{2N} = \hat{G}_{2B}^{-1} \odot \hat{Q}_2 \)

where \( \hat{\kappa}_{2B} \) and \( \hat{\kappa}_{2N} \) are the basic variable and the non-base variable of \( \hat{\kappa}_2 \), respectively. \( \hat{c}_{2B} \) and \( \hat{c}_{2N} \) are the corresponding parts with the basic and non-base aspect of \( \hat{\zeta}_2 \), respectively. \( \hat{G}_{2N} \) is the sub-matrix of \( \hat{G}_2 \) except \( \hat{G}_{2B} \).

**Proof:** With the block matrices defined above, the constraint condition of (31) can be written as:

\[
[\hat{G}_{2B}, \hat{G}_{2N}] |[\hat{\kappa}_{2B}, \hat{\kappa}_{2N}]^T = \hat{Q}_2
\]

where \( \hat{\kappa}_{2B} = \hat{c}_{2B} \odot (\hat{Q}_2 - \hat{G}_{2N} \odot \hat{\kappa}_{2N}) \). Substituting \( \hat{\kappa}_{2B} \) into (32), one can obtain

\[
J = < \hat{c}_{2B} | \hat{\kappa}_{2B} > + < \hat{c}_{2N} | \hat{\kappa}_{2N} >
\]

\[
= < \hat{c}_{2B} | \hat{G}_{2B}^{-1} \odot (\hat{Q}_2 - \hat{G}_{2N} \odot \hat{\kappa}_{2N}) > + < \hat{c}_{2N} | \hat{\kappa}_{2N} >
\]

\[
= < \hat{c}_{2B} | \hat{G}_{2B}^{-1} \odot \hat{Q}_2 > - < \hat{\lambda}_N | \hat{\kappa}_{2N} >
\]

where \( \hat{\lambda}_N = \hat{c}_{2B} \odot \hat{G}_{2B}^{-1} \odot \hat{G}_{2N} - \hat{c}_{2N} \) is the checking number.

**Remark 5:** If all checking numbers of the non-base variables are less than or equal to zero, basic solution corresponding to \( \hat{G}_{2B} \) will be the optimal solution of programming problem (31). If the checking number is zero and the corresponding \( \hat{Q}_2 < 0 \), there exists no optimal solution of (31). If some checking numbers and the corresponding \( \hat{Q}_2 \) are both larger than zero, the non-base variable corresponding to the positive checking number will become the basic variable. Continuous circulation is implemented until the first condition is satisfied, and one can obtain the final optimal solution of programming problem (31).

**V. Simulation Results**

To verify the effectiveness of the proposed NFTSM controller (17), simulations have been carried out using the rigid spacecraft system governed by (8). A TSM controller [15] is also simulated for comparison. It is noted that the presence of the sign function in (17) will result in chattering phenomenon. To solve this problem, the hyperbolic tangent function is used as substitution of the sign function.

It is assumed that the leader spacecraft is in circular orbit with 42240 km. The control objective of the follower spacecraft is keeping the same attitude and distance with the leader. The initial relative attitude and position of the follower spacecraft are chosen as \( p_e(0) = [-20, -10, -10]^T \text{m} \), \( q_e(0) = [0.6245, 0.5, 0.5196, 0.3]^T \), \( p_c = [0, 0, 0]^T \text{m} \), \( q_c = [1, 0, 0, 0]^T \), \( \omega_c(0) = [0, 0, 0]^T \text{rad/s} \). The external disturbance force and torque are \( u_d = [0.06 + \ldots] \).
0.03\sin(0.6t), 0.05 + 0.04\sin(0.9t), 0.04 + 0.01\sin(0.5t)]^T \text{N}
and \( \tau_d = [0.000002 + 0.0005\sin(0.8t), 0.00003 + 0.0003\sin(0.5t), 0.00001 + 0.0007\sin(0.3t)]^T \text{Nm} \). The nominal mass and inertia are \( m_0 = 100\text{kg} \) and \( J_0 = \text{diag}(18, 18, 24) \text{kgm}^2 \) while the actual ones are \( m = 95\text{kg} \) and \( J = \text{diag}(17, 17, 22) \text{kgm}^2 \). The control parameters are set as: \( \alpha = 0.67, p_1 = 0.6, p_2 = 0.67, \delta = 0.05 + 0.001, K = \text{diag}(20, 20, 20) + \epsilon \text{diag}(05, 05, 05) + \epsilon \text{diag}(01, 01, 01), \alpha_1 = \alpha_2 = \bar{b} = \text{diag}(05, 05, 05) + \epsilon \text{diag}(2, 2, 2) \).

Figs. 2 and 3 represent the time responses of relative position and attitude errors, where the relative position error driven by TSM controller converges to within 78s. In contrast, the proposed controller performs well because convergence rate and accuracy are both considered in the sliding mode and controller design. Thus, the relative position error driven by the proposed controller converges within 22s. In Fig. 3, the relative attitude in proposed controller converges faster than that in TSM. Note that the TSM controller leads to more transient oscillations in attitude and angular velocity responses, as shown in Figs. 3 and 4. It is noted that singularity phenomenon is eliminated in the proposed controller.

Based on the proposed controller, the following actuator configuration (see Fig. 5) is employed to test the proposed NSPI control allocation scheme using the improved simplex method. It is noted that #01-#16 denote corresponding thrusters in Fig. 5.

Figs. 6-9 show the practical outputs of each actuator in CTC. As observed, pair-mounted actuators can provide symmetrical thrusts. The feasible solution can be found in pseudo inverse method beyond thrust limitation 5N in CTC, where the negative values can also be offered by the thruster from the other direction. Compared with the conventional PI method, the improved simplex method can achieve control allocation requirements successfully despite control force limitations.
verse method are eliminated by the improved simplex method, which ensures that all the practical actuator outputs are subject to the limitation. Finally, numerical simulations to evaluate the overall performances for non-singularity, fast tracking, high accuracy, uncertainty resistance, finite-time stability have verified the effectiveness of the proposed method.

REFERENCES


VI. CONCLUSION

In this paper, the finite-time controller design and allocation problem of mechanical systems are studied. Based on the relative integrated dynamics, a fast terminal sliding mode is proposed using switching strategy. In addition, considering external disturbances and system uncertainties, a non-singular finite-time control law is proposed without the unwinding problem. Furthermore, disadvantages of traditional pseudo inverse method are eliminated by the improved simplex method,

Fig. 7. Control force of thrusters #05–08.

Fig. 8. Control force of thrusters #09–12.

Fig. 9. Control force of thrusters #13–16.
Detection of Defective Bolts from Rotational Ultrasonic Scans Using Convolutional Neural Networks

Duje Medak¹, Fran Milković¹ Luka Posilović¹, Marko Subašić¹, Marko Budimir², Sven Lončarić¹

¹University of Zagreb, Faculty of Electrical Engineering and Computing, Zagreb, Croatia
²INETEC Institute for Nuclear Technology, Zagreb, Croatia

Email: duje.medak@fer.hr

Abstract—Bolts are one of the primary components used when constructing complex systems such as power plants, factories, railways, and similar. Due to constant stress over the years, various types of defects can appear inside bolts, making the overall structure unsafe. Detection of defective bolts can be done by employing a non-destructive material evaluation technique, such as ultrasonic testing (UT). However, the amount of data acquired during the inspection is often large, so the analysis, nowadays performed manually, lasts a long time. In this work, we propose a method based on a convolutional neural network (CNN) to classify ultrasonic scans and detect defective bolts. We propose a novel representation of the ultrasonic B-scans that we call rotational B-scans. By transforming the original database of B-scans into this novel representation, the number of images displaying a defect increases. This balances the dataset, decreases the dataset variance, and makes the training of a deep convolutional neural network significantly easier. We tested many existing architectures and based on our findings we designed a custom encoder-decoder-based classifier. Our model outperformed all the other tested models and reached an area under the receiver operating characteristic curve (AUC-ROC) of 97.4%.

Index Terms—image processing, image analysis, ultrasonic testing, nondestructive testing, automated flaw detection

I. INTRODUCTION

Ensuring the safety and proper functioning of a system includes continuous monitoring of its critical parts. Bolts are one of the commonly used parts when creating complex structures such as transportation systems, oil and gas pipelines, various power plants including the nuclear power plants, and similar. It is necessary to ensure that the installed bolts are undamaged and can perform their normal function. This is done by inspecting each individual bolt periodically, detecting the defective ones, and planning their replacement. Inspection of bolts is performed by using non-destructive evaluation (NDE) techniques. NDE is used to examine objects of various types, sizes, shapes, and materials to determine the presence of discontinuities, such as defects or to determine other material characteristics [1], [2]. UT is a NDE technique, commonly used due to its simplicity and ability to inspect deeper parts of the material. It works by transmitting ultrasonic waves through the material and analyzing the received reflected energy. The ultrasonic waves reflect from the discontinuities inside the material, such as defects.

The received ultrasonic energy is usually displayed in the form of an A-scan, B-scan, or C-scan. A-scan shows the amplitude of the received ultrasonic energy as a function of time, and it is obtained when the probe transmits and receives one ultrasonic pulse. During the inspection, the probe is moved while transmitting and receiving ultrasound waves, so several A-scans are obtained. A sequence of A-scans can be stacked together into a new representation called a B-scan. This ultrasonic image is a neat way to speed up the data analysis, since images are easier to interpret than sequences of A-scans. Finally, a C-scan representation is also commonly used to get a rough idea about the locations of potential defects. This representation is formed by stacking several B-scans and performing top-view projection. Some information is lost in this process, but the inspectors who look at the data perform this projection for various depths of interest, and it allows them to quickly inspect some areas where the defects often appear. An illustration of the bolt inspection is shown on the left side of Fig. 1. The middle image in Fig. 1, shows a sequence of B-scan images that are obtained during ultrasonic scanning, and a C-scan that is created by projecting columns of B-scans. The right image shows what the sequence of B-scans looks like, after applying the new sampling strategy proposed in this work.

Ultrasonic scans also contain many signals created by the reflection of the ultrasonic waves from the boundaries of the inspected components, for example, the bolt’s head, or the joint between the bolt’s head and body. This can be seen in Fig. 2. The defect detection (manual or automated) is done by looking at the sequence of B-scans and determining whether a signal that appeared in a B-scan is caused by the geometry of the inspected component or by the reflection from a defect. A high variance between the B-scans being analyzed makes the inspection more difficult, as it increases the amount of signals that are suspicious and need to be classified. Sometimes it is very difficult to distinguish between the non-defective and defective signals, so years of training and experience are needed to perform this task reliably. Besides being non-trivial, the inspection of ultrasonic testing data is also very time-consuming. Some structures can have thousands of bolts.
Early works for automated analysis of UT data relied mostly on signal processing techniques such as wavelet transform and classification of features using support vector machines (SVM) [4], [5] or an artificial neural network (ANN) [6]. More recent works for automated analysis of NDE data employ deep learning techniques to detect various discontinuities such as cracks inside the material [7], [8], corrosion [9], [10], and welding defects [11]–[13]. While there is a rapid development of methods for automated analysis of various UT data, a thorough literature overview did not yield any publications dealing with the automated analysis of UT bolt scans. The UT data acquired during bolt inspections is, to the best of our knowledge, still analyzed manually [14], [15]. There are a few somewhat related works for automated inspection of screws and bolts, like automated visual inspection after manufacturing, or detection of bolt loosening. In [16] the authors proposed an automated screw quality inspection based on a simple and resource-efficient CNN. They developed their model by using a dataset of 8200 images taken after the screw manufacturing. Their approach achieves an accuracy of 95.1% while being several orders of magnitude faster compared to the quality inspection performed by humans. In [17] the authors test four popular CNN-based classifiers to detect different types of defective screws. In their experiments, InceptionV3 architecture was better at detecting screw head defects, while MobileNetV2 was better for the detection of defects inside the screw’s body. Another example of defective screw detection is presented in [18]. The authors develop a robust CNN-based method that allows them to perform defect detection in a real industrial process. They achieve this by training the model with position-independent images, which enables their quick and automated acquisition. Their approach, similar to the one proposed in this work, relies on existing CNN architectures, data augmentation [19], and transfer learning [20]. Some other tasks related to the automated NDE testing of bolts include detection of bolt loosening and detection of missing bolts. In [21], the authors proposed a probabilistic neural network to predict torque values, which can be used to detect bolt loosening. In [22] the authors propose hierarchical object detection to automatically analyze images of moving trains and detect if something is out of the ordinary. The authors evaluated their approach on multiple tasks such as detection of defective components, missing components, or component displacement. One of the tasks discussed in this work is the detection of missing bolts from the train’s b-plates.
As seen in the given overview of related work, many researchers are trying to develop more reliable and faster ways of ensuring the quality of components and their proper functionality. In this work, we focus on automated analysis of UT scans of bolts. We propose a novel representation of the data that enables the development of CNN-based methods for automated analysis. We test many of the popular architectures and based on our findings propose a novel image classification architecture. The details of our approach are given in the following section.

II. METHODOLOGY

Fig. 3 shows a high-level illustration of our approach. We first collect a standard dataset of B-scans by scanning 27 bolts. We then employ the strategy described in subsection II-A to transform the collected data into a database of rotational scans. The resulting scans are more suitable for training the CNN classifier. We train several existing architectures and based on our findings design a novel CNN classifier. The trained model can be used to analyze new rotational bolt scans and detect defective bolts.

A. Rotational ultrasonic B-scans

The original dataset was obtained by scanning 27 bolts in total - 18 bolts containing various artificially created defects inside the material, and 9 non-defective bolts. During the scanning, a robotic manipulator moves the probe as shown in the left image in Fig. 1. Each time one cross-section of the material is scanned (one B-scan is acquired), the probe is shifted a bit and the process is repeated. For one bolt, 26 standard B-scans are collected. Some of those B-scans display a border of the bolt, where the defects do not appear as often as in the B-scans displaying the center of a bolt. This means that some scans are more important than others, due to a higher probability of containing a defect. The scanning was done with a probe transmitting the ultrasound waves at a 0° angle. During the data acquisition, the initial position of the robotic manipulator is not always perfectly calibrated, so the collected data is not centered. Human experts can easily find the areas of interest and perform analysis, even when inspecting the shifted data. However, the creation of the rotational data requires centered data as input. We propose a novel approach to accomplish this.

Our approach relies on the appearance of characteristic circles in C-scans that are formed by the regular and circular shape of the bolt’s head and body. An example of these characteristic circles can be seen in Fig. 4. We process and analyze the C-scans and detect the centers of bolts. First, we artificially stretch the data to obtain a squared image. We also decrease the original resolution of C-scans from $650 \times 650$ pixels to $40 \times 40$ pixels in order to reduce the complexity of the following steps. To ease the detection of the circle created by the bolt’s head, we invert the pixel values and perform histogram equalization. We then perform Hough transform to detect circles from the detected edges. We filter out only the circles with radius sizes between 15 and 25 pixels. The described process is shown in Fig. 4.

Once we have obtained the estimates of a bolt’s center and radius, we align the data by cropping it equally on all sides from the circle’s center. We do not perform cropping on downsampled data, but instead, we transform the calculated center and radius back to the original size. We then crop the original B-scans to the appropriate width and remove the B-scans that do not display the bolt (B-scans that are outside the circle on the y-axis). Once the data is centered, we resize the B-scans to $96 \times 96$ pixels, as this is enough to perform further analysis. For each bolt, we generate a new tensor where each of the B-scans is replicated multiple times to match the B-scan’s width. The resulting tensor then has a dimension of $96 \times 96 \times 96$. An example of a C-scan created from such a tensor is shown in Fig. 5. Each pixel of this C-scan is obtained by projection, but we do have access to actual column values from which the projection was calculated. We can thus sample this data using a different strategy and create a novel rotational dataset. We want all of our new scans to look similar, so the ideal sampling strategy is to create new B-scans that all pass through the center of the bolt. The benefit of the proposed approach is that all the scans have the same importance and an equal probability of containing a defect. The proposed sampling strategy is also illustrated in the rightmost image of Fig. 1. We determine the points of sampling by choosing an initial angle, the final angle, and the step. In our case, we sample the data starting with the horizontal line shown in Fig. 5, and then rotating the line 18 times by 20°. This means we generate 18 rotational scans for one bolt. Since the step is 20°, the other half of created scans corresponds to the horizontally flipped first half. We do this on purpose as a form of data
augmentation, which is useful when training the classification model. The locations of defects inside the bolts is known from the bolts’ blueprints, since the defects were artificially created. This information was used to manually annotate all B-scans displaying a defect. Created annotations were later used to determine whether a rotational B-scan contains a defect or not. Our final rotational dataset contains 274 non-defective and 266 defective scans. We randomly create 5 splits of the data to evaluate the models for automated analysis. In every fold, the bolts used in the test subset are never used during the training. We also make sure we do not use the same bolt in the test set twice across folds. The table I shows the number of training and testing images per fold. As it can be seen from the table, the dataset is balanced as the number of non-defective scans is roughly equal to the number of defective scans.

**B. Convolutional Neural Networks for ultrasonic scan classification**

While the rotational B-scans can be used to aid human inspectors with the data analysis, the main motivation for creating such a dataset was to enable the development of a method for automated detection of defective bolts. To achieve this, we train several popular CNN-based image classification models. After the training, the models can be used to detect scans containing defects and thus automatically detect defective bolts. During our experiments, we noticed that the smaller and simpler architectures outperform complex ones. Therefore, we developed a custom image classification network. The proposed architecture consists of an encoder-decoder part, followed by an additional encoder and dense layers for classification. The architecture was inspired by the model proposed in [23] that showed great results in UT image analysis. The architecture consists of five encoding convolutional layers with 32 filters of size 4 and stride 2. A Leaky ReLU activation function is used after the convolutions. The decoder part is created by transposed convolutions and skip connections that add feature maps from the encoder to the corresponding feature maps of the decoder. Finally, another encoder is used to extract more features. This encoder uses 64 filters of size 3, and MaxPooling layers to perform down-sampling. The outputs of the encoder are flattened and passed through 3 dense layers with the dimensions 256, 128, and 2.

**C. Experimental setup**

We trained ten existing image classification models and one custom model proposed in this work. We train all the

<table>
<thead>
<tr>
<th>Fold</th>
<th>Normal Training</th>
<th>Defective Training</th>
<th>Normal Testing</th>
<th>Defective Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>236</td>
<td>232</td>
<td>38</td>
<td>34</td>
</tr>
<tr>
<td>2</td>
<td>234</td>
<td>234</td>
<td>40</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>238</td>
<td>230</td>
<td>36</td>
<td>36</td>
</tr>
<tr>
<td>4</td>
<td>232</td>
<td>236</td>
<td>42</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>234</td>
<td>234</td>
<td>40</td>
<td>32</td>
</tr>
</tbody>
</table>
models with cross-entropy loss using an SGD optimizer with a learning rate of 0.001, and a momentum of 0.9. We use batch size 16 and perform 500 training steps per epoch. We use 20% of the training data for validation to reduce the learning rate on plateaus, and early stop the training. The learning rate is reduced by 10 times if there is no improvement in validation loss for more than 10 epochs. We stop the training after 20 consecutive epochs without validation loss improvement and use the best-saved model for final testing.

We normalize the data by subtracting \((0.485, 0.456, 0.406)\) and dividing with \((0.229, 0.224, 0.225)\). We randomly crop patches from the input images as a form of data augmentation. The crops are between 80% and 100% of the image size, with aspect ratios randomly chosen from the range \([0.9, 1.1]\). We also perform random perspective transformation with a distortion scale of 0.1 and 50% probability. The images are resized to \(224 \times 224\) pixels before feeding them to a model. The test and validation data was resized and normalized, but not augmented. All the models were implemented in the PyTorch library [24] and the augmentation was done using the Torchvision package [25].

### III. RESULTS AND DISCUSSION

The created dataset has many advantages compared to the original representation, especially when trying to develop an automated approach for the detection of defective bolts. The number of defective and non-defective examples is balanced in the rotational dataset, and the defects are often seen better. Furthermore, the rotational scans are more similar to each other. To quantify this, we calculated per-pixel variance values of the original dataset and compared them to the values calculated from the rotational dataset. The results are shown in Fig. 6. By using the proposed rotational representation, the average variance is decreased from \(6.53 \times 10^{-5}\) to \(3.11 \times 10^{-3}\) and the maximum variance is decreased from \(0.14\) to \(0.08\). As expected, the variance in the rotational dataset is mostly high around the areas where defects appear, while in the original dataset it is more spread out. Finding the region of interest, where the defects might appear, is thus much easier on rotational scans.

We used the created rotational dataset to train popular image classification architectures. The results are shown in Table II.

---

![Fig. 6. A per-pixel variance of the original dataset (left) and the rotational dataset (right).](image)

---

The models were evaluated using the area under the receiver operating characteristic curve (AUC-ROC). The best results are achieved by our custom model, which is also the simplest model among the tested ones, with only 1.27 million trainable parameters. The best performance among the existing CNN architecture was achieved by the VGG16 model [26]. To further test the developed models, we also plotted attention scores of VGG for one non-defective and one defective input image. This is shown in Fig. 7. We used GradCAM [27] to visualize where the model focuses when making a decision. In non-defective images, the focus is on the inside body of the bolt. In defective images, the attention scores are high around the defect’s signal, but also around the characteristic signals. This confirms the rotational dataset allows for a better model focus.

### IV. CONCLUSION

In this work, we propose a novel approach for the detection of defective bolts from ultrasonic testing images. First, we propose a novel data representation that is created by resampling the collected data. We refer to this novel representation as rotational B-scans and demonstrate many benefits of its usage. After creating the dataset of rotational B-scans, we use it to train CNN-based image classifiers. Besides testing many existing CNN architectures, we also designed a simple custom model. This model outperformed all the other tested models and reached an AUC-ROC score of 97.4%. The results
prove that the proposed method can be used reliably to assist human inspectors with the detection of defective bolts or to even automate this process in the near future. The method should be further tested on a larger dataset consisting of data captured with different phased array probes.

ACKNOWLEDGMENT

This research was co-funded by the European Union through the European Regional Development Fund, under the grant KK.01.2.1.01.0151 (Smart UT).

REFERENCES


Abstract—Non-expert users are increasingly affected by the decisions of systems that rely on machine learning (ML), yet it is often difficult for these users to understand the predictions of ML models. In this paper, we propose a web-based platform to evaluate explainable AI (XAI) for non-experts in the context of time series forecasting, focusing on energy price predictions as an exemplary use case. The XAI methods we consider include local feature importance and counterfactual explanations. The platform relies on gamification to encourage user engagement. Our research objective is to evaluate the effectiveness of these different approaches from the perspective of non-expert understanding of machine learning models.

Index Terms—XAI, global, local, counterfactual explanation

I. INTRODUCTION

Machine learning (ML) and Artificial Intelligence (AI) has demonstrated effectiveness across a range of domains such as finance [1], medicine [2], social media [3] and autonomous driving [4], [5]. However, for those who interact with AI, the lack of transparency and ability to understand their actions may affect their trust in such systems. This is especially noticeable when the system is making high-stake decisions [6] where human understanding of those decisions is vital. In the medical industry [2], for example, diagnosing a condition early can be critical. In the banking sector, AI-enabled banking systems may use credit scores to decide if a loan request is granted, and suitable explanations are expected if a loan request is denied. Explanations are required not just by researchers, but may also be required by regulatory obligations that are becoming more prevalent, such as the General Data Protection Regulation (GDPR) [7] which requires that the underlying principles for algorithmic predictions be transparent, and that the rationale behind those predictions be explainable, if necessary.

One common categorisation of machine learning models, within this context, is white box and black box models, with the former models decision-making considered transparent, and the latter less so. Further, XAI approaches can be characterised as producing global or local explanations, or as being post-hoc or ante-hoc [8]. That is, global explanations provide a higher level insight into a model while local explanations provide individual instance-specific explanations. Post-hoc methods provide explanations for an already trained model whilst ante-hoc methods produce an additional component (to the trained model) to aid explanations.

In [9], the authors divided users of an AI system into three classes based on their expertise: AI beginners, domain experts, and AI experts. This is based on the belief that distinct groups have different requirements. While AI experts may find utility in global explanations of how machine learning models function, beginners and (non-AI) domain experts may benefit more from local explanations about the relationship between specific inputs and outputs. Although explanation has been found to increase understanding of ML systems for a wide range of audiences [10] [11], non-experts are chosen as an important but under-represented class of stakeholder within XAI research [12].

Gamification is the incorporation of game elements into systems or activities for the purpose of motivating and engaging users [13]. Gameplay data from users can be logged and analyzed, with the aim of improving the underlying system. For example, [27] utilised a type of gamification called games with a purpose (GWAP) [14] to evaluate XAI at scale. The authors concentrated on explaining deep learning models that have been built for image recognition, and claim it is the first time that GWAP has been used within XAI research. In [15], the authors replaced the game design with user objectives in XAI planning, and added money as an incentive. They connect the user’s objective (or goal) to payment via a bonus that grows in proportion to the goal value attained, which gives an incentive to develop a successful strategy.

In this paper, we design and develop an interactive XAI system for non-experts. As an application domain we use an ML model that performs energy price prediction. We design two interactive user interfaces. One interface is intended for users in a control group, whilst another interface is intended for users in a treatment group. Our proposed gamification uses a simple game to allow users to experience how the value of different features influences predictions of the underlying ML model. A simple score mechanism is used to inform a user how their own predictions differ from the system’s predictions so that, with repeated play, a user can gain a better understanding of the underlying ML model.

The rest of the paper is as follows: Section 2 introduces the dataset and gamification. Section 3 discusses global, local and counterfactual explanations. Section 4 describes the system design. Finally, we provide a brief summary and our plans for the future in the conclusion.
II. PRELIMINARIES

A. Dataset Description

The dataset used in this study is Octopus Energy import prices\(^1\) for the London region during a two-year period beginning midnight on 1 January 2018 and ending at midnight on 31 December 2019. The original dataset is a time series dataset containing 34,993 instances at 30 minute intervals, with the current electricity price with and without tax\(^2\). In order to use standard regression algorithms we pre-process the dataset to extract the following set of time-based features: year, month, day, hour, day of week, is weekend. In addition to those, weather and carbon intensity features are also added to the dataset to improve the predictive performance of the model. Carbon intensity data comes from carbon intensity forecast in National Grid ESO\(^3\), which is described in terms of a 96+ hour forecast of CO\(_2\) emissions per kWh of consumed electricity. Meteomatics\(^4\) provides temperatures in degrees Celsius for the London area and is updated every half-hour.

To select a regression model we first evaluate several common regression models to identify a performant model for our XAI experiment. We emphasise that we are not interested in identifying a state-of-the-art model for this task, but that any reasonably well-performing model will suffice. We trained several regression models, using the final month for evaluation. Mean squared error (MSE) is the evaluation metric used which measures the mean squared difference between the predictions and the ground truth. Table I provides the performance of the top three regression models we tested, in terms of MSE, with the Random Forest model achieving the best performance. Furthermore, the performance from these models improved when carbon intensity and temperature were added as extra features, with the MSE for the best performing model reducing from 6.33 to 3.31.

<table>
<thead>
<tr>
<th>Regression Model</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random Forest</td>
<td>3.31</td>
</tr>
<tr>
<td>Extra Trees</td>
<td>3.56</td>
</tr>
<tr>
<td>Histogram-based Gradient Boosting Regression Tree</td>
<td>3.75</td>
</tr>
</tbody>
</table>

B. Gamification

The inclusion of a gamification component in our XAI system provides a metric for us to evaluate user performance. In our study, we seek to measure the understanding that a non-expert has of an underlying ML model. To aid this, we design a game where the user receives a score based on their understanding. This Score will be received during gameplay. Despite the fact that there are presently a variety of explanatory approaches for black box models, such as LIME and SHAP, quantitative assessment methodologies for XAI are presently lacking. There is currently no agreement on how best to assess explanations [16]. In one example, [16] evaluated explanations in terms of application-ground, human-ground and function-ground. However, no precise objective evaluation techniques are provided. Several researchers have validated the efficacy of explanations by conducting user studies. In [17], over 200 participants were asked to evaluate an XAI interface. The study deployed an assessment measure called the Explanation Satisfaction Scale to measure satisfaction after providing explanation. However, this assessment approach is static and not based on incentives, while the subjective nature of user-reported satisfaction makes it difficult to judge the real usefulness of explanations. Our gamification and scoring mechanism aims to overcome this limitation.

Our game interface asks the user to estimate the relationship between a model’s predicted price \(X\) and a hypothetical estimated price \(Y\), given a specific input instance, with a limited number of qualitative options for the user to choose from. Those options are much less than, less than, similar to, greater than, much greater than. The reason we ask for a qualitative estimation is to place less cognitive load on the (non-expert) user. Each option is associated with a qualitative numeric value from \((-2, -1, \ldots, 2)\) as shown in Table II. The absolute difference between the qualitative numeric value of the model’s prediction and the user’s choice, normalised by the maximum absolute difference (i.e. \(2 - (-2) = 4\)), is then taken as the user’s score. Thus, scores are taken from \(\{0, 1, \ldots, 4\}\) with 4 being the best score and 0 being the worst.

For example, if the user estimates that \(X\) is much greater than \(Y\), and \(X\) is actually less than \(Y\), then the user will be awarded the score of \(4 - |2 - (-1)| = 1\). Scores thus provide a way to measure user understanding of the underlying ML model, and follow the common-sense understanding in games that a higher score is better.

<table>
<thead>
<tr>
<th>Option</th>
<th>Semantics</th>
<th>Numeric value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Much less than</td>
<td>(X &lt; Y-20)</td>
<td>-2</td>
</tr>
<tr>
<td>Less than</td>
<td>(Y-20 \leq X &lt; Y-5)</td>
<td>-1</td>
</tr>
<tr>
<td>Similar to</td>
<td>(Y-5 \leq X &lt; Y+5)</td>
<td>0</td>
</tr>
<tr>
<td>Greater than</td>
<td>(Y+5 &lt; X \leq Y+20)</td>
<td>1</td>
</tr>
<tr>
<td>Much greater than</td>
<td>(Y+20 &lt; X)</td>
<td>2</td>
</tr>
</tbody>
</table>

III. EXPLANATIONS FOR BLACK BOX MODELS

Broadly speaking, machine learning models can be divided into two categories related to their interpretability, namely, black box models and white box models. Black box models models are difficult to understand on their own [18], and as a result, their is a greater need for explanation. On the other hand, white box models [18] are regarded as interpretable by design, and hence are easier to explain.

According to the results of our experiments, the Random Forest model, which may be considered a black box model,
performs the best at our task. A random forest model is a collection of decision trees, each trained on a different subset of the data with random subsets of the features. As a result, viewing each tree (of the typically large number) is not a viable explanation approach for non-experts.

Some explanation methods that do not consider the specifics of the machine learning model model, but only the inputs and outputs, are collectively referred to as model-agnostic explanations [19]. Model-agnostic explanation can be separated into global and local approaches, depending on the aim of interpretation.

A. Global Explanation

To interpret the model’s global output, the model must be trained to understand the algorithm and the data. This level of interpretability refers to the model’s decision-making process in relation to the full feature space and model structure. The most popular global approach is Permutation Feature Importance (PFI) [20]. The PFI is used to determine which feature has the most influence on the prediction. The PFI selection approach evaluates a model’s performance after eliminating each unique feature and replacing it with random noise. Individual feature importance may thus be directly compared, and a quantitative threshold can be utilized to determine feature inclusion. In Fig. 1, the rows towards the top are the most important features, and those towards the bottom matter least. Thus, hour is regarded as the most important feature, whereas day is seen as the least important.

### Table III

<table>
<thead>
<tr>
<th>Feature</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>hour</td>
<td>1.6310 ± 0.0432</td>
</tr>
<tr>
<td>temperature</td>
<td>0.0679 ± 0.0035</td>
</tr>
<tr>
<td>day of week</td>
<td>0.0036 ± 0.0014</td>
</tr>
<tr>
<td>day of week_num</td>
<td>0.0005 ± 0.0005</td>
</tr>
<tr>
<td>lsWeekend</td>
<td>0.0003 ± 0.0003</td>
</tr>
<tr>
<td>month</td>
<td>0 ± 0.0000</td>
</tr>
<tr>
<td>year</td>
<td>0 ± 0.0000</td>
</tr>
<tr>
<td>day</td>
<td>-0.0011 ± 0.0010</td>
</tr>
</tbody>
</table>

![Table III](image)

Fig. 1. Global explanation for the energy price dataset

B. Local Explanation

[21] points out that local explanation offers a tailored explanation that is focused on the particulars of each instance. It provides a thorough explanation of how a machine learning model may provide precise predictions about the features effecting a specific prediction. The authors studied the global and local interpretability of machine learning in Type 2 diabetes screening, and found that characteristics such as age, gender and body mass index (BMI) contributed significantly to global explanation. However, for a specific patient they found that depression, smoking status or physical health had a significant impact on the development of Type 2 diabetes in that patient.

In our work, we use SHapley Additive exPlanations (SHAP) [22] to generate the local explanation. SHAP can find the feature importance which can interpret the predictions of any machine learning classifier or regressor. SHAP includes two sub-methods: KernelSHAP and TreeSHAP. We utilise TreeSHAP to explain our Random Tree regressor.

To illustrate, we use the SHAP method to provide a local explanation for one instance of our dataset. Table III depicts the feature values of an instance that we want to investigate.

### Table III

<table>
<thead>
<tr>
<th>Feature</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>hour</td>
<td>1.6310 ± 0.0432</td>
</tr>
<tr>
<td>temperature</td>
<td>0.0679 ± 0.0035</td>
</tr>
<tr>
<td>day of week</td>
<td>0.0036 ± 0.0014</td>
</tr>
<tr>
<td>day of week_num</td>
<td>0.0005 ± 0.0005</td>
</tr>
<tr>
<td>lsWeekend</td>
<td>0.0003 ± 0.0003</td>
</tr>
<tr>
<td>month</td>
<td>0 ± 0.0000</td>
</tr>
<tr>
<td>year</td>
<td>0 ± 0.0000</td>
</tr>
<tr>
<td>day</td>
<td>-0.0011 ± 0.0010</td>
</tr>
</tbody>
</table>

According to Fig. 2, we can not only see the influence trend of each feature, but we can also see how the features contribute to a single prediction. Shap values deconstruct a prediction to demonstrate the impact of each feature. The resulting price is £7.52, while the base value is £12.24. The feature values that cause greater predicted values are highlighted in pink, and their visual size indicates the magnitude of the feature’s effect. Blue represents feature values that decrease the predicted value.

![Fig. 2. Local Explanation](image)

In this scenario, the local explanation concludes that features “temperature” and “carbon intensity” have a positive influence on the prediction, while “hour”, “year”, and “day” have a negative influence. Based on this plot, we could generate the following textual local explanation in a pop-up window: “temperature” and “carbon intensity” are the two most inferential features for energy price in this case”.

C. Counterfactual Explanation

[23] discussed that the end-user may not be particularly interested in why a certain prediction was obtained and which features of the input led to the prediction. Instead, they may be more interested in understanding the changes that can be made to obtain other predictions. Counterfactual explanations primarily address the issue of how the prediction will change if a certain change in the features of the input occurs. It compares the user’s expectations with the actual predicted outcomes, and provides suggestions about how to change feature values in order to alter prediction results. In some cases, counterfactual explanation is more intuitive and useful within the local explanation category. Furthermore, some psychologists [24] [25] have demonstrated that counterfactuals elicit causal reasoning in humans.

Counterfactual explanations are commonly selected based on some measure of proximity to the original input. By restricting which features to focus on (and what range of
values they can take), specific counterfactual instances can be found to better meet a user’s expectation.

We can use the same instance in Table III as an example to generate explanations for using DiCE [26], which generates counterfactual explanations of machine learning models. In Fig. 3, the top row represents the feature values of the explainee datapoint and the bottom rows represent two counterfactual datapoints. In these two rows, features with “-” mean that these features have the same values as the explainee datapoint. Accordingly, the features with values present are those which are different from the explainee datapoint’s features.

![Fig. 3. Counterfactual Explanation](image)

We implemented and integrated DiCE as part of the interface and Fig. 3 is part of the interface, where a pop-up arrow will point to the feature values that have influenced the ML to produce the alternative outcomes.

IV. SYSTEM DESIGN

A. System Architecture

Fig. 4 illustrates how the architecture of our proposed system. The system involves three main components. They are front-end, back-end and the dataset. The front-end is end-user facing, which receives user input to be transmitted to the back-end via an API, which then performs tasks such as prediction and explanation generation. The dataset component is used purely to train the underlying ML model which the user is seeking to understand.

![Fig. 4. System architecture](image)

B. Interface Design

The interface has three panels. They are the visualisation, game interface and game board panel. Fig. 5(a) depicts the visualisation panel. The visualisation is presented to the user with a stepped line chart. In our system it displays the energy tariff price in the future (e.g., one year from now), with a point at every half-hour interval. Additionally, for exploration, the top of this panel displays a week calendar picker to allow users to change calendars week by week. Fig. 5(b) illustrates two sliders for the carbon intensity and temperature feature values. The default carbon intensity is set to 150 gCO2/kWh, and the default temperature is set to 20 °C. The scale of the carbon intensity is from 0 gCO2/kWh to 300 gCO2/kWh and the scale of temperature is from 0°C to 40°C.

![Fig. 5. Visualisation and feature slider controls.](image)

The game interface panel is depicted in Fig. 6(a). The purpose of this panel is for users to explore the system interactively. Fig. 6(b) shows the game board displaying the results of each round of the game. There are five rounds for each player to play. The game board is divided into six columns for treatment group participants. They are as follows:

- **day/time**: it shows the date and time of each round.
- **User’s answer**: it show the result inferred from the visualisation screen.
- **Whether the user requests explanation or not**: it shows whether an explanation is required (i.e, yes or no)
- **User’s answer (second)**: it shows the result inferred after viewing the textual explanation.
- **Correct answer**: it shows the true answer.
- **Score**: it shows the score received for each round.

![Fig. 6. Game interface and game board panel](image)

C. Interaction Design and Gamification for Local Explanation

The treatment group user interface uses gamification for local explanation. That is, each round of games is about one specific datapoint where data feature values are given, and through gamification, a user is tasked with the predicting the prediction of the underlying ML model for that datapoint. Although gamification used in this paper does not provide counterfactual examples, through the process of playing games, a user can obtain comparisons between different instances. The user can analyse the correlation between feature input values and their predictions and could learn from each round of play.
We provide a case study of the gamification interface for the treatment group. Fig. 7 demonstrates the whole interface for treatment group participants.

Step 1: The game interface is populated by the feature values of the datapoint whose price is to be predicted.

- **datetime**: 07:00–07:30 on Wednesday 1 January 2025 (this should occur in the future)
- **carbon intensity**: 112 gCO₂/kWh.
- **temperature**: 2.4°C.

Step 2: The task is for the user to utilise explanations in order to determine if the a hypothetical price is close to what the underlying model would predict given the features. For this reason, each round of the game includes a hypothetical price. To utilise the visualisation, the user can modify the carbon and temperature features via the sliders (e.g., move temperate slider to 2.4°C, and the carbon intensity slider to 112 gCO₂/kWh). They can then inspect the chart for time-related trends by changing e.g. the month or year. For example, users may inspect:

- Prices at 07:00am-07:30am every day.
- Prices every Wednesday.
- Prices on the first day of every month.
- Prices in May every year.

After completing all operations and analysis, the user triggers the button to refresh the visualisation with the matching parameters.

Step 3: The system asks the following question: **What do you think the relationship is between the model true predicted value and the hypothetical estimated price?**

A user needs to select a **Radio button** (corresponding to the qualitative options introduced previously) as an answer based on their analysis of the visualisation. The system will then display a message to tell the user whether their choice is correct.

Step 4: Users are asked further questions **Do you require some explanation?**. User can select either **Show** or **No, Finish**. If the user selects **Show**, a local explanation will be displayed. If the user selects **No, Finish**, the game will move on to the next round. After selecting **Show**, the user is prompted to select again by the sentence “Please choose again:”. Once done, the user can **Submit**. The game repeated in this way 5 times.

Step 5: All recorded information is shown in the **game board** with **Score** being calculated by the system.

In the game board panel, two critical pieces of information will be recorded by the system. Firstly, it is important for users in the treatment group to determine whether the user requests the textual explanation or not. This information may indicate whether the user felt confident in their understanding of the visualisation. Secondly, the users score is important as a metric of their understanding and will aid in further analysis of user performance.

V. CONCLUSION

In this paper, we proposed a web-based XAI system consisting of visual and textual components using gamification to measure non-expert user understanding of a ML prediction model. For illustrative purposes, we used home energy price prediction as the application. The proposed system is flexible in that it supports not only various explanation modalities, but also different types of explanations, i.e., local and counterfactual explanations. In future work, we will conduct real-world user experiments using the proposed system.

REFERENCES


Abstract—Home burglaries and house fires occur daily and bring hardships to the victims that suffered from these unfortunate events. Early detection of these situations allows for rapid responses and should always be a highly demanded feature amongst all homeowners. In this study, we propose an AI assistance framework specifically tailored to detect common household features using the home’s pre-existing surveillance cameras. This framework will allow homeowners to implement an extra layer of security on top of their existing hardware. We developed a prototype that detects abnormalities of fires, people, vehicles, and house pets. To the best knowledge of the authors, this is the first application of its kind to collectively monitor such a wide variety of activities to provide a holistic view of the state of home. We have verified the sustainability of the framework to monitor the home in real time.

Keywords—Surveillance camera, AI Assistance Framework, Abnormality Detection, Convolutional Neural Network (CNN)

I. INTRODUCTION

Home burglary is still prevalent in this day and age, and unfortunately this event occurs daily in the UK. According to the office for national statistics, the UK saw a total of 258,594 home burglaries in 2021 alone [5]. This means on average a burglary in the UK occurs every 30 seconds and found that the average loss from a burglary was around £2000. Not only does this have a financial impact on the victims, but also leaves a big emotional and mental toll on them. Being able to add an extra layer of security will help ease the mind of a homeowner and allow them to be trusting in their home’s security.

House fires also cause major issues to families and homeowners. Whenever this event occurs, it causes huge financial loss to the homeowner, and unfortunately may even cause life loss. According to government statistics [6], the UK alone saw 61,910 house fires in the 2020/21 year. This means that roughly 7 house fires occurred every hour. Response to these fires must be taken quickly, in order to stop the fires from engulfing the home [7].

By utilizing the home’s pre-existing surveillance systems, we can add an extra layer of security and help the people feel at ease in their own homes.

The rest of the paper is organized as follows: section II gives the related work; details of the proposed method are in section III; datasets used in the paper are briefed in section IV followed by results and discussions in section V; the paper is concluded in section VI.

II. RELATED WORK

Trying to visually detect fire and smoke has been studied and has found to be successful. Frizzi et al [1] introduced a two-part model in order to detect fire and smoke in videos. Where part one consisted of four convolutional layers and two max pool layers, and part two consisted of two fully connected layers and the output layer. However, because of the resource intensive of this model for prediction, a sliding window of size 12x12 pixels was applied on the last feature map which increases the speed of predictions.

One challenge to train a satisfactory fire and smoke prediction model is the limited available data. A study done by Namazov and CHO [3] indicated that a good fire and smoke image classification model can be achieved with limited data by employing tactics using data augmentation to squeeze performance. The adaptive piecewise linear function also provided a notable performance improvement in their model without increasing the number of learnable parameters.

A study conducted by Howard [2] presented that using a variety of Image transformations on an image, can greatly improve the accuracy of CNN-based image classification. It is because each transformation will be a new unseen image and so this can greatly increase our training data set. Howard [2] tested five models on the dataset with no transformations applied, and then applied the following transformations to the dataset: 9 crops, 2 flips, 3 scales and 3 views. The models were tested again using the new transformed dataset, which turns a single image to 162 images to predict on. All models showed an increase in accuracy with the lowest increase in accuracy being 11.7%.

Accuracy assessments as the most integral part in image classification procedure, have been reported as the most employed and successful approach for evaluation classifications in a survey conducted by Lu and Weng [4]. The assessment based on error matrices can find the weakest links in the chain, helping the author identify the section that needs the most work that will be critical for the accuracy of the model.

In regards the size of models, a review conducted by Pham et al [16] summarized over 200 publications discussing video-based human action recognition using different deep learning models and algorithms. The results show that the larger CNN such as VGG and ResNet often produced higher accuracy scores and expected these networks to be exploited in the human detection field in the future. One drawback of using these larger networks is the number of layers they contain. This leads to believe a more specialized and lighter model could have greater success, as large networks could negatively impact our framework due to its more resource demands to run these models in real-time with sufficient framerate [17].

From all the related work stated so far, a couple of features can be employed to advance our framework further. Data augmentation must take place, as this has proven to be extremely successful in reference [2] and [3]. We will also be using accuracy assessments and error matrices to successfully evaluate our framework based on reference [4].
Whilst Frizzi et al. [1] had a good model and achieved success in their project, they did not have such a varied multi-class model and so we will be building upon their two-part model approach, in order to find a successful model.

III. METHODOLOGY

This section will outline the proposed framework with specific components, the data pre-processing, and the training and validation functions that will be used.

A. Proposed AI Assistandce Framework

The proposed framework consists of three components, the CCTV system, the Onboard processing, prediction and decision logics, and the Alarm and remote alerting module shown as Fig. 1. The CCTV system contains multiple cameras which entirely cover the house range and garden footpath. The video footage from each camera is fed into the Onboard logic where the model is located. The inputs can be from live CCTV systems, or recorded footages for offline analysis. To maximize the compatibility of the proposed framework, the Onboard logic is capable of capturing videos played on the screen by any software. The CNN model is sufficiently lightweight to run using onboard processing instead of relying on cloud resources and outputs one of five events, which then be fed to the Decision logic for prediction. Based on the output from the model, corresponding Alarm messages with video streams will be transferred to alert relevant users.

Fig. 2 illustrates the overall architecture of the CNN model from Fig. 1, which is a classic classification model combining the convolution layers, max pooling layers, dropout, flatten and dense layers. It starts with three convolutional layers followed by ReLU activation. The convolutional kernels of these three layers are respectively $4 \times 4$, $7 \times 7$ and $5 \times 5$ and the numbers are 64, 128 and 256. We then apply a max pooling layer with a 3x3 pool size and feed the convolutional feature-maps into a Dropout layer. The final convolutional layer has 256 $4 \times 4$ kernels followed by a $1 \times 1$ max pooling and Flatten.

The detail of the model is summarized in Table 1. Conv2D layers will take a multi-channel input (the RGB video frames or multi-channel feature-maps) and apply the filters across all the channels and then sum the result. It creates a convolutional kernel which helps to produce a tensor of outputs. A kernel is a convolutional matrix which is used for many different image processing techniques. This gives us a monochrome convolved image [11, 12].

<table>
<thead>
<tr>
<th>Layer (type)</th>
<th>Output Shape</th>
<th>Param #</th>
</tr>
</thead>
<tbody>
<tr>
<td>conv2d_40 (Conv2D)</td>
<td>(None, 199, 199, 64)</td>
<td>3136</td>
</tr>
<tr>
<td>conv2d_41 (Conv2D)</td>
<td>(None, 97, 97, 128)</td>
<td>401536</td>
</tr>
<tr>
<td>conv2d_42 (Conv2D)</td>
<td>(None, 47, 47, 256)</td>
<td>819456</td>
</tr>
<tr>
<td>max_pooling2d_29 (MaxPooling2D)</td>
<td>(None, 23, 23, 256)</td>
<td>0</td>
</tr>
<tr>
<td>dropout_52 (Dropout)</td>
<td>(None, 23, 23, 256)</td>
<td>0</td>
</tr>
<tr>
<td>conv2d_43 (Conv2D)</td>
<td>(None, 10, 10, 256)</td>
<td>1048832</td>
</tr>
<tr>
<td>max_pooling2d_30 (MaxPooling2D)</td>
<td>(None, 5, 5, 256)</td>
<td>0</td>
</tr>
<tr>
<td>flatten_10 (Flatten)</td>
<td>(None, 6400)</td>
<td>0</td>
</tr>
<tr>
<td>dropout_53 (Dropout)</td>
<td>(None, 6400)</td>
<td>0</td>
</tr>
<tr>
<td>dense_30 (Dense)</td>
<td>(None, 500)</td>
<td>3200500</td>
</tr>
<tr>
<td>dropout_54 (Dropout)</td>
<td>(None, 500)</td>
<td>0</td>
</tr>
<tr>
<td>dense_31 (Dense)</td>
<td>(None, 100)</td>
<td>50100</td>
</tr>
<tr>
<td>dropout_55 (Dropout)</td>
<td>(None, 100)</td>
<td>0</td>
</tr>
<tr>
<td>dense_32 (Dense)</td>
<td>(None, 5)</td>
<td>505</td>
</tr>
</tbody>
</table>

Total params: 5,524,065
Trainable params: 5,524,065
Non-trainable params: 0

The MaxPooling2D layers will down sample the input along the spatial dimensions. It does this by taking the maximum value over the input window for each channel of inputs.

The Flatten layer simply flattens the input. We will also utilize Dropout layers during each step of the training to improve the generalization and robustness of the model. To finish the model, Dense layers are applied, which connects all the previous layer nodes to the nodes in this layer, each node will provide an output to the next layer.

The network then enters the final phase, where it has three steps of Dropouts followed by Dense layers with ReLU activation again. The input-output nodes of the Dense layers are respectively 6400-500, 500-100 and 100-5, which shown as the example of Fig. 3.
The outputs of the CNN model classify the input images into five different events, that is Vehicle, Pets, Fire/Smoke, Human and Default, which respectively correspond to different objects. Notably, the Default images do not include any above objects shown as Fig. 4. The classification results enable the decision logic shown as Table 2 (i.e., Lookup table) to identify the Risk Level for Alarm by sending the messages with video streams to users.

<table>
<thead>
<tr>
<th></th>
<th>Vehicle</th>
<th>Pets</th>
<th>Fire/Smoke</th>
<th>Human</th>
<th>Risk Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>x</td>
<td>1</td>
<td>x</td>
<td></td>
<td>High Risk</td>
</tr>
<tr>
<td>x</td>
<td>x</td>
<td>0</td>
<td>1</td>
<td></td>
<td>Medium Risk</td>
</tr>
<tr>
<td>1</td>
<td>x</td>
<td>0</td>
<td>0</td>
<td></td>
<td>Low Risk</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>Low Risk</td>
</tr>
</tbody>
</table>

1 Vehicle presents for a prolonged duration.
2 Pets appear at least once in a given period.

Both proposed frameworks will be manually activated and deactivated by users. For example, the houseowner activates the system after driving the only vehicle to work and deactivate it when they return. During the system working period, the detected objects from multiple cameras such as vehicles and humans are not belonging to here while the pets may get lost if undetected for a given duration. Table 2 uses x, 0 and 1 to respectively represent irrelevant (don’t care about their presence), negative and positive of detection. Vehicle presents for a prolonged duration will be set as 1 while Pets do not appear at least once in a given period will be marked as 0. The situations not included in Table 2 are non-Risk situation.

We define three Risk Levels, i.e., High, Medium and Low, to generate and transfer messages along with video streams to users. The Fire/Smoke is taken as the first priority for High Risk alarm, and the Human appearance is the Medium Risk which could be treated as an intruder. Vehicle mis-parking and Pets lost are the Low Risk situations. It’s worthy noting that users can set the priorities at their own preferences.

B. Training

The Adam optimizer has been chosen during the compilation of the model. This is a stochastic gradient descent method and combines the best properties of AdaGrad and RMSProp to provide the most optimized algorithm that can handle noisy problems or sparse gradients. The optimizer defaults to a learning rate of 0.001.

The Categorical Cross-entropy class is selected as the loss function for multi-class classification. Categorical accuracy is also mostly used to show the performances of multi-classification, which calculates how often predictions match the one-hot labels.

C. Validation

The validation set is used on the test data which will give the categorical accuracy of the test data after every epoch. This will help when fine-tuning the model, but this might also negatively affect the model as it can directly influence our design decisions.

IV. DATA

This section will outline the datasets that have been recognized as viable and will be used to train the model and will include ways in which we further enhance these datasets.

A. COCO Dataset

COCO (Common Objects in Context) [8] is a large-scale object detection dataset, containing over 330,000 images with over 80 object categories. It was created with the goal of aiding advanced image recognition, which is now the facto dataset to benchmark real-time object detection performance. This dataset will be used to train the model to recognize “Default” scenes. This will not include any of the other objects (i.e., Vehicle, Pets, Fire/Smoke and Human) we are training the model on shown as Fig. 4.

B. Open Image Dataset V6

The Google Open Image Dataset [9] is very similar to the COCO dataset. It has 1.9 million images which makes it the largest existing dataset with object annotations. It also includes detailed annotations that include object relations, such as “woman playing guitar” or “beer on table” and has over 19,900 classes. This dataset also comes with a package which can be installed using pip. It allows for users to easily download the images using their API in the command line interface with following classes to train the model: Vehicle (Car and Motorcycle as Fig. 5), Human (Fig. 6) and Pets (Cat and Dog as Fig. 7).
C. Fire/Smoke Dataset

Byoungjun Kim and Joonwhoan Lee created this Fire/Smoke dataset [10] as they were working on creating a deep learning-based fire detection method that automatically will generate a set of useful features after training. Due to this they needed to gather a large amount of training data and so this dataset was created for the training and testing of their experiment and published it as an asset for future computer vision-based fire detection.

It consists of diverse still images and video clips and includes and enhances the data from other well-known public datasets (in which they do not elaborate which datasets were included) shown as Fig. 8. We use this dataset for Fire/Smoke recognition.

D. Data Pre-Processing

Monochrome (greyscale) images prove to be an issue when reading in the images. When these images are located, discarding completely or using OpenCV to expand the channel to match those of RGB images. We have decided to make these images usable due to the fact that some surveillance systems are going to record in grayscale, and so it will benefit our model to train some features in grayscale, so that it can learn and recognize these features when no color is presented, allowing our model to be useful in these systems.

All images will be resizing to 400×400, as the input data needs to be uniform to feed into our neural network. The larger the fixed size, the more pixels, and data the model will have to work with, ultimately increasing training time. Images smaller than this will be resized to match this size, and images larger than this size can either be resized down to this size or the bordering pixels can be cropped to ensure that key details can still be seen. Therefore, not to crop the features we are looking for as this could negatively impact the learning of the model, and so we have opted to simply resize the image down.

The entire dataset will have a 70/10/20 split, with 70% of the samples being used to train the model, 10% being used to test and 20% used to validate.

E. Data Augmentation

The Data augmentation contains flipping and color manipulations. Original image will be flipped symmetrically and followed by randomly manipulate the contrast, brightness, and color of both images in a random order, using a number between 0.5 and 1.5, by using PIL [13][14] similar to Howard, A.G. [2]. These manipulations generate an extra 3 images from the original image used, that should try to cover some variation to help the model learn invariance to changes in these properties. Fig. 10 shows an example of this data augmentation taking place. The rest of the samples are

V. RESULTS AND DISCUSSION

In this section, we will break down the model’s accuracy and loss per epoch during training. The Confusion Matrix with a heatmap visual aid will be used to analyze and also, the classification performance of Accuracy, Precision, Recall and F1 score will be provided in Table 3.

<table>
<thead>
<tr>
<th>Class</th>
<th>N. Truth</th>
<th>N. Classified</th>
<th>Prec.</th>
<th>Recall</th>
<th>F1 score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicles</td>
<td>373</td>
<td>419</td>
<td>0.70</td>
<td>0.79</td>
<td>0.74</td>
</tr>
<tr>
<td>Pets</td>
<td>310</td>
<td>422</td>
<td>0.49</td>
<td>0.66</td>
<td>0.56</td>
</tr>
<tr>
<td>Default</td>
<td>222</td>
<td>216</td>
<td>0.58</td>
<td>0.57</td>
<td>0.58</td>
</tr>
<tr>
<td>Fire/Smoke</td>
<td>2394</td>
<td>2120</td>
<td>0.99</td>
<td>0.88</td>
<td>0.93</td>
</tr>
<tr>
<td>Human</td>
<td>105</td>
<td>227</td>
<td>0.31</td>
<td>0.68</td>
<td>0.43</td>
</tr>
</tbody>
</table>

TP: 2802
Overall Accuracy: 82.31%

1 The number of Truth; 2 The number of Classified; 3 Precision.
The N. Truth represent the actual number of labels in this category while the N. Classified shows the number of cases classified as belonging to this category.

The model reached 87% accuracy on epoch 14 (shown as Fig. 11), however looking at the F1 score of the classes, the Default, Human and the Pets classes have quite a low score. When comparing this to the N. Truth of these classes, it may be due to the lack of data for these classes. Where there are only 105 correct samples for the Human class, and 222 for the Default class.

Figure 11. Training Accuracy and Validation Accuracy Learning Curve per epoch

Fig. 12 is the training and validation loss graph. As there is generally no large gap between training accuracy and validation accuracy, it can be concluded that our model is not overfitting in these 15 epochs. Also, there is sufficient representative data present. However, due to the curve has not started to plateau, the final potential of the model has not been achieved.

Figure 12. Training Loss and Validation Loss Learning Curve per epoch

Another observation based on the Fig. 12 loss graph is that the slight irregularities in the validation curve could suggest that the validation dataset does not provide sufficient information to be able to evaluate the ability of the model to generalize. The steps taken to improve and counter this issue should be, to increase the size of the dataset used so as to allow for a bigger validation set or another method could be to balance the split of the dataset, which could negatively impact the training set and so the best approach is to collect a larger dataset.

The Confusion Matrix with a heatmap visual aid is shown as Fig. 13. According to Confusion Matrix, a general horizontal trend across the matrix told that the model is predicting somewhat correctly. Fig. 14 shows an example of correct classification.

The colour of the Fire/Smoke class presents much deeper than the other classes in Confusion Matrix. It is due to a class-imbalanced dataset, where the Fire/Smoke has a much greater amount of image samples than the other classes. This has been done as this is potentially a fatal occurrence, and so we must make sure that this specific feature is well trained for.

Another observation made is that the model predicted around 130 Pets image samples (Cat and Dog) as Fire/Smoke. Fig. 15 shows an example of where the model predicted an image to be Fire/Smoke, but it was a part of the Pets class.

Figure 13. Confusion Matrix with a Heatmap Visual Aid [15]

Figure 14. Example Vehicle Image where the model predicted correctly

Figure 15. Example where the model predicted an image to be Fire/Smoke, but it was a part of the Pets class.
One final observation made, is that the model often predicted the Human class images as Fire/Smoke. One obvious reason for this is that there are fire fighters and other personnel present in many of the Fire/Smoke images shown as Fig. 16. This could have skewed the learning of the Fire/Smoke class and potentially have overlapped some features with the human class. These kinds of images output High Risk alarm due to the presence of Fire/Smoke no matter containing other objects.

VI. CONCLUSION

In this study, we proposed a model to detect abnormal scenes at homes particularly when the home owner is away. This includes humans, home pets, vehicles, fire and smoke. When the model is integrated into home surveillance systems, an extra layer of security for the homeowner can be added. The image samples extracted from public datasets include many of the objects that the model tries to classify together. We manually clean the data to make sure that the irrelevant objects did not influence training. Though the learning curve shows that the final potential of the model has not been achieved, the results still provide correct classification.

The limitations of our proposed framework are: (a) The training samples taken from public datasets instead of particular CCTV footages might affect the performance on CCTV system; (b) Lack of comparing performance with other baseline CNN models; (c) The Alarm and remote alerting cannot recognize complicated situations. As a result, our future works will collect more CCTV footages to retrain the model and compare with other baseline model. Also, developing the CNN model with intelligent alarm system for human behaviour prediction and situation awareness.

ACKNOWLEDGMENTS

This work is partially supported by Vital Sign Point of Care Analysis Tool project funded by the Innovative UK (KTP/12053).

REFERENCES

Deep CNN based droplet deposition segmentation for spray distribution assessment

Tao Chen  
School of Computer Science and Electronic Engineering  
University of Essex, Colchester, U.K.  

tc17339@essex.ac.uk

Yanhua Meng  
School of Mechanical Engineering,  
Anyang Institute of Technology, Anyang, China  
yanhua.meng@outlook.com

Jinya Su  
Department of Computing Science  
University of Aberdeen, Aberdeen, U.K.  
jinya.su@abdn.ac.uk

Cunjia Liu  
Department of Aeronautical and Automotive Engineering  
Loughborough University, Loughborough, U.K.  
C.Liu5@lboro.ac.uk

Abstract—Pesticides have been widely used in the cultivation of crops to enhance their production, however, incorrect application of pesticides will result in yield loss, product waste, environmental pollution among many others. Therefore, timely evaluating spray distribution of intelligent sprayers plays a pivotal role in the appropriate delivery of pesticides to the crop. The exiting approaches based on water-sensitive paper (WSP) either involve a relatively tedious and labor-intensive procedure, or have a high requirement on WSP image taking. So in this study we aim to conduct spray distribution assessment in the field based on mobile devices. To this end, the key issue of droplet deposition segmentation under natural imaging environments is addressed. WSPs with food dye droplets are first collected in the field by mobile phones. Then an image dataset on droplet deposition segmentation is created via thresholding approach with human supervision. Then four popular deep convolutional neural network (CNN) based segmentation algorithms are applied for droplet deposition segmentation so that spray distribution can be assessed. Comparative experiments show that UNeXt network is the best one in consideration of accuracy, inference time and network size.

Index Terms—Convolutional neural network (CNN); Droplet segmentation; Pesticide spray analysis; Precision agriculture; Semantic segmentation; Water sensitive paper.

I. INTRODUCTION

An increasing world population, projected to be 9 billion by 2050, is placing an unprecedented demand on agriculture, almost 70% more food demand increase. This global challenge is even severer by considering the scarcity of the arable land and natural resources, climate change and the societal demand for shrinking agriculture’s environmental footprint for its sustainable development [1], [2]. Crops, however, are threatened by various stresses (e.g., abiotic stresses, pathogens and pests) in their life-cycle, which if managed inappropriately will lead to yield loss and quality degradation, posing serious threats on food security. In particular, it is estimated that about 30% of the crop loss worldwide is due to the adverse effects of weeds, diseases and insect pests annually [3]. Two approaches are generally available to attenuate the side effects of these issues including chemical control and genetic resistance [4]. Although there is much progress in genetic resistance, at present, chemical control, i.e., the application of pesticides is still the dominant approach (e.g. representing a $40 billion yearly budget) to enhance crop production, particularly in developing countries and regions [5]. In this context, it is significant that the right amount of pesticides should be sprayed on the target crop areas of interest. This is because, on the one hand, excessive uses of pesticides may leave residues in the agricultural products along with ecological tainting (e.g. increasing the likelihood of ground war contamination) and a high cost, generating significantly economical, environmental and social burdens [6]; on the other hand, insufficient doses of pesticides may result in areas of the harvest fields that are not protected, lessening crop productivity.

Therefore, it is highly desirable to develop an automated spray distribution assessment system so that spraying variables and parameters (e.g., nozzle type, droplet size, flow rate, operational parameters of spraying systems such as flight height and velocity in UAVs based sprayers) of agricultural sprayers can be evaluated and optimised in a short time window for a better spray distribution [3]. Various efforts have been made for this purpose in the past two decades including imaging and no-imaging based approaches. It is noted that only imaging based approaches are briefly introduced in this work due to the lack of space.

One notable work for this end is the DepositScan framework [7] proposed in 2011, which is made of the DepositScan software program and a handheld card scanner. In this framework, the WSPs are first collected, and scanned by the handheld business card scanner, which can obtain the image pixel information due to the fixed width of scanner and the scanner resolution (i.e., dots per inch or dpi). Then the scanned images can be analysed by the open-access ImageJ software.
in the DepositScan program. As a result, different key metrics for spray distribution can thus be calculated including spray coverage, number of droplets in unit area and droplet size distribution. It is noted, however, that this scanning based framework is inconvenient for field applications due to the relative tedious and labor-intensive procedure therein.

Recent efforts have been made to develop field devices or applications for onsite spray distribution assessment instead of offline indoor processing. For example, [8] designs an intelligent vision sensor node for WSP image collection, which can adapt to the changes of light intensity in the environment. Then a watershed segmentation algorithm is applied to separate the droplets in the image. However, the system involves an elevated cost, constraining its wide adoptions.

Alternatively, there are also several smartphone based imaging approaches, which rely on smartphones for image collection and algorithm implementation. This approach is conceived to be promptly accessible, and portable to the fields. In this context, SnapCard [9] is the first pesticide spray coverage tool running on a smartphone, which only supports the coverage area of the WSPs. DropCard with DropScope is a commercial smartphone application which relies on an external water-card reader under restricted card sizes. In 2021, DropLeaf [5] is proposed, where image analysis pipeline is developed to analyze the WSP images taken carefully by the smartphone. The pros and cons of the three smartphone based applications for pesticide spray assessment are summarized in Table 1 of [5]. It should be noted that the state-of-the-art DropLeaf has a high requirement on WSP image taking.

Therefore, the main aim of the study is to develop a droplet deposition segmentation system so that images of WSPs taken in the field conditions by various mobile devices can be automatically segmented, so that the spray distribution of various sprayers can be evaluated quickly in the field conditions. To this end, WSPs are first collected at different locations of the citrus wogan structure and under different spraying parameter settings. Then images of the WSPs are taken in the field conditions by different mobile devices. Considering the high workload involved in manual droplet annotation, thresholding method is then used to segment the droplets from the WSP background for the purpose of image labelling, where the threshold is manually chosen by human experts. Suitable regions of interest of the segmented WSP images are chosen to construct the training image dataset. On this basis, various deep CNN based segmentation approaches are compared to identify the most suitable one for droplet deposition segmentation. Some spray distribution metrics can thus be calculated based on the segmented images. To be more exact, the main contributions are summarized as below.

1) **Dataset:** An image dataset for droplet deposition segmentation is generated by taking images of WSPs and labelling via manually choosing thresholds and region of interest (ROI) cropping.

2) **Algorithm:** Various deep CNN based segmentation algorithms are compared for droplet deposition segmentation with good performance.

## II. Materials and Methods

In this section, materials and methods relevant to the study are introduced such as WSP image collection, image labelling and labelled dataset generation, droplet deposition segmentation and metric calculation for spray distribution assessment. To ease the understanding of this study, the flowchart of the developed framework is displayed in Fig 1.

![Flowchart of the developed framework including WSP image collection, image labelling, labelled dataset generation, droplet deposition segmentation and metric calculation for spray distribution.](image)

### A. WSP image collection

In the experiment, instead of directly spraying pesticide, red food dye is adopted due to its low price and color differences from the WSP background. WSPs are placed at three different positions of the citrus crop. At the same time, different operation parameters are adopted in the experiments so that different droplet distributions can be obtained.

Upon spraying droplets being collected by the WSPs, WSPs are placed above the printed reference calibration checkerboard, where the black and white squares have a fixed size of 1 cm. Then images are taken by different mobile phones under different illumination conditions, where the imaging rule is summarized as below: (1) the phone (with down-facing camera) is placed above the WSPs so that the WSPs are in the center of the image; (2) the phone is close enough to the WSPs to guarantee a high spatial resolution but meanwhile also make sure the images have no blur; (3) at least one white square should be visible in the image so that the pixel physical size can be determined. Some exemplary WSP images with different spray distributions are displayed in Fig. 2.

### B. Methods

In recent years, image segmentation has been undergoing a significant progress, especially with the popularity of deep learning methods. In this study, we also aim to exploit deep CNN learning technique to segment the spraying droplets from WSP background. Deep learning methods, especially supervised deep learning methods, are generally data-hungry, and there is no such a dataset which focuses on droplet deposition segmentation. Hence, we create a dataset in which annotation is done by thresholding color images in Hue-Saturation-Value (HSV) color space. Image labelling and labelled dataset generation, droplet deposition segmentation...
Figure 2. Exemplary WSP images with different spray distributions taken by different mobile phones under different illumination variations.

and metrics calculation for spray distribution assessment are detailed as below.

1) Image labelling and dataset generation: In order to build the deep CNN based droplet deposition segmentation model, a labelled dataset should be first generated for this supervision learning task. Considering the high workload in manually labelling the droplets one by one in WSPs, an image labelling algorithm in Algorithm 1 is proposed.

In this approach, a suitable grayscale image is first obtained by the RGB color bands or color space transformation (e.g. HSV) of the original RGB image or different vegetation indices [10]. Then the thresholding approach is adopted for the grayscale images, where the threshold is manually chosen for each sample WSP so that the labelling errors can be minimized. In addition, due to the substantial illumination changes in practical field conditions, the thresholding approach may be only valid for some part of the WSPs even various thresholds are chosen for different WSPs. So the correctly segmented regions are cropped out by abandoning the wrongly segmented areas.

Until this step, we already have the dataset and its labels for training a deep network, but the images so far have various image sizes (e.g. length and width), which is unable to feed to different deep networks for a fair comparison. Therefore, after image labelling, we further process the images to a fixed image size, which is 128 \times 128 pixels in our setup. In particular, given a labelled image, a sliding window of size of 128 \times 128 starts from the top-left corner of the image and crop the corresponding regions of size 128 \times 128. Meanwhile, the corresponding label images are also cropped in the same way as its original label images so that image pairs are generated including source image and label image. Then the sliding window is moved with the stride of 10 \times 10 pixels in width and height directions until it reaches the bottom-right corner of the image. To this end, our dataset consists of 3600 images where 80% are used for training, 10% for validation and the remaining 10% for testing. In Fig. 3, four selected exemplary images and their labels are shown. Note that these images are taken under various lighting conditions.

Algorithm 1 Semi-automatic labelling for dataset generation

| Input: WSP images. |
| Output: Labelled dataset for model construction. |
| **Step 1:** Do RGB band separation or color space transformation (e.g. HSV) to obtain a suitable grayscale image; |
| **Step 2:** Separate the droplet depositions from the background via thresholding approach with manually chosen threshold for each WSP; |
| **Step 3:** Crop the regions with good droplet segmentation performance by visual inspection; |
| **Step 4:** Construct the labelled dataset by generating a large number of image patches with a fixed size of 128 \times 128 from the images in Step 3. |

Figure 3. Exemplary image pairs including source RGB image (first row) and label image (second row) from our created dataset.

2) Droplet segmentation: Upon generating the labelled dataset for algorithm construction, we now consider the problem of droplet segmentation. In order to identify the most suitable model for droplet segmentation, we evaluate four state-of-the-art semantic segmentation networks including VGG16 [11], VGG16+CBAM (Convolutional Block Attention Module) [12], Unet [13], [14] and UNeXt [15] on our dataset. Specifically, UNet and UNeXt are particularly-designed networks for medical images segmentation by using a few annotated data. In addition, VGG network is a well-known architecture and has shown significant performance in many computer vision tasks. In this study, we also explore the combination of VGG and an attention network CBAM to improve the representation of the features. The CBAM module is used to enhance the feature representation using two attention networks in channel and spatial axis, respectively. We insert the CBAM module into the second last layer of VGG16. The dimension of the features remain the same before and after the addition of the CBAM module. In order to obtain the droplet segmentation results, we modify the last layers of the aforementioned four networks, letting them generate two feature maps including background and foreground with the same size of input image. Their qualitative and quantitative comparison results will be analysed in section III.
**Performance evaluation:** To evaluate the performance of segmentation networks, we use metrics including Recall, Precision, F1-Score, Average precision score (AP), and Intersection over Union (IOU), where a higher value indicates better performance. In particular, the Recall and Precision are defined as below:

\[
Recall = \frac{T_p}{T_p + F_n}, \quad Precision = \frac{T_p}{T_p + F_p}
\]

where \(T_p\) represents the number of true positive, \(F_n\) means the number of false negative, and \(F_p\) is the number of false positive.

F1-score, the harmonic mean of precision and recall, is able to consider recall and precision simultaneously. It is useful for imbalanced dataset and is calculated as:

\[
F_1\text{-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}.
\]

AP summarizes a precision-recall curve as the weighted mean of precisions achieved at each threshold, with the increase in recall from the previous threshold used as the weight. AP is defined as below:

\[
AP = \sum_n (Recall_n - Recall_{n-1}) \times \text{Precision}_n.
\]

where \(\text{Precision}_n\) and \(Recall_n\) are the precision and recall at the \(n\)th threshold.

IOU is the ratio of area of overlap/intersection over area of union, where a value closer to 1 means better performance.

\[
IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}}.
\]

3) **Metrics for spray distribution:** After droplet segmentation, the next step would be computing the metrics evaluating the spray performance. There are generally three metrics quantifying the spray distribution including spray coverage (SP), number of droplets per unit area (NDUA, droplets/cm²), droplet diameter distribution. In particular, SP refers to the relative zone occupied by the droplets in the Region Of Interest (ROI), which is calculated as below Eq. 5

\[
SP = \frac{\text{NoPixel by droplets}}{\text{NoPixel in ROI}}.
\]

where \(\text{NoPixel}\) denotes the No. of pixels.

Droplet diameter distribution is usually quantified by \(D_{V0.1}, D_{V0.5}, D_{V0.9}\), which represent the distribution of the droplet diameters such that droplets with a diameter smaller than \(D_{V0.1}, D_{V0.5}, D_{V0.9}\) compose 10%, 50% and 90% of the total liquid volume, respectively [7]. It is noted that in the calculation of the above metrics (except SP), the pixel resolution (e.g., the physical size of each pixel in the image) is vital, which is detailed as below.

In image analysis, the \(\text{Area}_i\), of the \(i\)th droplet can be calculated by the number of pixels occupied by the droplet multiplying the physical (instead of pixel) area of each pixel, which is given by

\[
\text{Area}_i = \text{NoPixel}_i \times width_{px}^2.
\]

where \(\text{NoPixel}_i\) denotes the No. of pixels occupied by the \(i\)th droplet, and \(width_{px}\) refers to the physical size of each pixels (usually in the unit of \(\mu m\)), which can be calculated from the reference calibration checkerboard as below.

\[
width_{px} = \sqrt{\frac{\text{Area}_{ws}}{\text{NoPixel}_{ws}}}.
\]

where the \(\text{Area}_{ws}\) denotes the physical area of the black/white square in the reference checkerboard (a known constant via its fixed size) and \(\text{NoPixel}_{ws}\) refer to the No. of pixels occupied by one black/white square in the reference checkerboard. The detection of black/white squares for the \(width_{px}\) will be implemented in the future.

Upon calculating the \(\text{Area}_i\) of the \(i\)th droplet, the droplet diameter of the \(i\)th droplet can be derived from the formula of the circle area \(\text{Area}_{circle} = \pi \times \left(\frac{\text{diameter}}{2}\right)^2\), given by

\[
\text{Diameter}_i = 2 \times \frac{\text{Area}_i}{\pi}.
\]

Then the calculation of \(D_{V0.1}, D_{V0.5}, D_{V0.9}\) is followed by statically analysing the diameters of all droplets. It is noted that alternative/modified formula is also available in [7].

Finally, the calculation of NDUA is given as below

\[
\text{NDUA} = \frac{\text{No. of droplets}}{\text{Area}_{ROI}}.
\]

where \(\text{Area}_{ROI}\) denotes the area of ROI and the No. of droplets in a ROI can be automatically obtained by applying image analysis function on the droplet segmentation images.

III. RESULTS

This section presents the results including training details, test performance of different segmentation networks and spray coverage calculation.

A. Training details

The four networks are trained with a batch size of 16 and 200 iterations. The Adam optimiser is used for optimising VGG and VGG+CBAM with learning rate \(1e^{-4}\). For the training of Unet and Unext, their hyperparameters are identical as in [13] and [15]. The networks are tested on a Nvidia GTX 1080 Ti graphic card. The loss function used for training is semantic cross-entropy loss, and it takes approximately 5 hours to converge.

B. Droplet segmentation performance

In Table I, we present the evaluation results of four networks on testing set. As one can see, all selected networks can achieve an F1 score of more than 80%, showing a relatively good performance.

Inference time: In Table II, we also list the time of networks for processing one image in the feed-forward calculation. The VGG network has the lowest inference time: 6.6 milliseconds per image. The lapsed time between mentioned networks is short and therefore it can be used for real-life applications.

Network Parameters: The size of network is a key index when deploying in practical applications, particularly, using
deep CNN in embedded system. Furthermore, considering the analysis to be done in the smartphone or other portable devices instead of workstation, we need a light-weight model for this task. In the second row of Table II, we summarize the parameters of the aforementioned network. Both VGG and Unet networks have encoder-decoder like architecture, but Unet utilises the skip-connection, which can bring up global and local features to decoder part to construct more accurate performance. CBAM is a light-weight attention module, combining with VGG increases 24MB parameters compared with VGG only.

Table I
Quantitative evaluation results of four CNNs

<table>
<thead>
<tr>
<th>Model</th>
<th>F1</th>
<th>Precision</th>
<th>Recall</th>
<th>AP</th>
<th>IOU</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG</td>
<td>0.817</td>
<td>0.739</td>
<td>0.93</td>
<td>0.817</td>
<td>0.695</td>
</tr>
<tr>
<td>VGG+CBAM</td>
<td>0.845</td>
<td>0.779</td>
<td>0.933</td>
<td>0.848</td>
<td>0.736</td>
</tr>
<tr>
<td>Unet</td>
<td>0.884</td>
<td>0.838</td>
<td>0.958</td>
<td>0.873</td>
<td>0.799</td>
</tr>
<tr>
<td>UNeXt</td>
<td>0.903</td>
<td>0.947</td>
<td>0.989</td>
<td>0.973</td>
<td>0.824</td>
</tr>
</tbody>
</table>

Table II
Inference time and network size

<table>
<thead>
<tr>
<th>Model</th>
<th>Time (second per frame)</th>
<th>Total size (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG</td>
<td>0.0066</td>
<td>229.3</td>
</tr>
<tr>
<td>VGG+CBAM</td>
<td>0.0068</td>
<td>249.34</td>
</tr>
<tr>
<td>Unet</td>
<td>0.0073</td>
<td>268.26</td>
</tr>
<tr>
<td>UNeXt</td>
<td>0.007</td>
<td>6.49</td>
</tr>
</tbody>
</table>

It is shown from Table. I and II that UNeXt network obtains the best performance in consideration of F1 score, Precision, Recall, AP, IOU, computation time and total size and so should be adopted in practical applications. This is mainly due to its improved design of network architecture, where convolutional networks are only used for learning low-dimensional features in early stage and tokenized Multi-layer perception (MLPs) to represent high-dimensional features by using few parameters compared with Convolutional network. Especially, this design demonstrates the ability to represent local dependencies that result in better segmentation performance in our test dataset.

Segmentation visualization: In Fig. 4, we present the qualitative segmentation results of the four CNNs, where the segmentation results of VGG, VGG+CBAM, Unet, and UNeXt are from second column to last column. There are five exemplary images with various illumination and spraying distributions chosen from our testing set. From the second column, VGG assigns some pixels to the background (smaller black holes in the white areas) which are false negative. These wrongly assigned pixels are mainly located in the spraying areas with unequally red color. With the auxiliary of CBAM, these false negative pixels have reduced slightly. From fourth column, we find that Unet can handle this problem better. But in the extremely case like the fifth source image, it encounters large areas of blurry droplets, VGG, VGG+CBAM, and Unet cannot overcome this issue properly. In the contrast, Unet can still recognise these pixels as droplets.

C. Spray coverage calculation

In this subsection, we report the SP index of WSPs based on the segmentation result of Unext. The whole process of SP calculation is illustrated in Fig. 5. In step 1, we manually crop the WSP from the background as the physical size of pixel is not required in SP calculation. It should be noted that the cropped images have different image sizes from the training samples and can not be directly feed to Unext. So in step 2, we resize the images to a larger square size $1024 \times 1024$ so that Unext can process them. In step 3, the fixed-size images are provided to Unext to obtain the masks of droplet. In SP calculation, the masks (white pixels) are treated as the droplet and the whole images are treated as the ROI. In the end, the spray distribution can simply be calculated via Eq. 5, which are 5.7% and 38.1% for WSP1 and WSP2, respectively.

IV. CONCLUSIONS AND FUTURE WORK

This work considers the challenges of spray distribution assessment via water sensitive paper images taken in field conditions. The main task under investigation is first transformed into droplet deposition segmentation under illumination changes, where droplets are labelled in a semi-automatic manner via manually tuned thresholds. Then four popular deep convolutional neural networks (CNNs) including VGG, VGG+CBAM, Unet and UNeXt are compared to identify the most suitable one for droplet deposition segmentation task. It is shown that UNeXt is the best one in consideration of accuracy performance, inference time and network size.

Although the performance of droplet deposition segmentation is satisfying and some metrics quantifying spraying distribution (e.g. spray coverage) are also obtained, there is still much room for further development in order that the developed system can be applied in real-life application for in-field spray distribution assessment. These aspects are summarized as below.

1) The dataset can be expanded to include more real-world field imaging conditions so that the semantic segmentation models become more robust and reliable.

2) More spraying distribution metrics (e.g. droplet number in unit area, droplet diameter distribution) can be calculated with the advert of pixel size information obtained from the reference checkerboard, and compared against the ground truth metrics obtained by the widely accepted approaches (e.g. Deposit Scan).

3) It would be interesting to see whether the developed framework can be translated to directly evaluating spraying distribution on crop leaves instead of water sensitive papers by using food dye of suitable concentration.

REFERENCES


Figure 4. Exemplary segmentation results: from left to right, RGB source image, VGG, VGG+CBAM, Unet, Unext.

Figure 5. Procedures of obtaining ROIs


Narrow gap TIG arc weld process monitoring

1st Nathan Gambrill
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, UK

2nd Michael Farnsworth
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, UK

3rd Rob Shaw
The Welding Institute Ltd
Abington, Cambridge, UK

4th Stefano Tedeschi
The Welding Institute Ltd
Abington, Cambridge, UK

5th Divya Tiwari
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, UK

6th Ashutosh Tiwari
Department of Automatic Control and Systems Engineering
The University of Sheffield
Sheffield, UK

Abstract—Joining processes play a critical role across several industries and manufacturing activities. One such activity is narrow gap TIG arc welding, present in a number of high value manufacturing domains the quality and outcomes are highly dependent on the operating parameters and the evolving state of the weld. Human supervision, and the application of vision systems to allow the operator to monitor the process as it is being undertaken is commonplace, however monitoring, characterisation and feedback in narrow gap use cases is less so. The work undertakes a comparative study into two different approaches to process monitoring of TIG arc welding. The first through more traditional machine vision (MV) methods, and the second through more modern deep neural network methods. Results show both methods developed are able to perform the role of process monitoring and defect detection with accuracy of 82% and 93% for the traditional and DNN methods respectively.

Index Terms—TIG Welding, Machine Vision, Deep Neural Networks, Process Monitoring

I. INTRODUCTION

Joining processes, in particular welding, play a critical role across several industries and production lines. These processes can be both automated or performed by manual operators and are highly dependent on the operating parameters and the evolving state of the weld as the process is undertaken. These include welding speed, arc current / voltage, laser power, defocused distances, shielding gas and root gap [1]. Tungsten Inert Gas (TIG) welding for narrow gap structures is one example joining process which requires stringent standards such as the ISO10042 [2] to ensure the best performance in its application. This is made more difficult given the nature of the gap and its limited spacing hindering any post-process reworking of defects. TIG is highly dependent on human supervision, and the application of vision systems to allow the operator to monitor the process as it is being undertaken are commonplace. Therefore, extending this to allow the use of recent advances in machine vision and learning for the process monitoring and defect classification of narrow gap TIG welding activities could provide several benefits. Process monitoring and control of welding through the use of machine vision however can be complicated through the high contrast and lighting conditions caused by the arc light which can obscure the melt pool and surrounding weld. The narrow gap itself also limits the sensor that can be positioned to observe the welding process. Nevertheless, advancements in machine vision, particularly in the field of deep learning offer the opportunity to provide real-time process monitoring and characterisation of the build which can then be used to provide feedback to automated or manual control of the evolving process.

The remainder of this paper explores two approaches to the real-time process monitoring of narrow gap TIG arc welding. The first is a look into traditional machine vision methods and how they can be utilised to monitor and perform defect classification of a narrow gap TIG welding process. The second is to then explore how current state of the art deep neural network (DNN) methods can be similarly applied to the same use case and then compared in terms of performance, exploring the drawbacks and benefits of both methods. Section 2 outlines the TIG arc welding use case and experimental setup used throughout the study and data collection process. Following this in section 3 we outline the motivation and methodology used to develop and evaluate the two different approaches, before ending with results and discussion in section 4.

II. TIG ARC WELDING USE CASE

The application of TIG arc welding can be found across a number of industries, in particular those within high value manufacturing such as aerospace. A more specific example can be found in pipeline manufacture where a multi-pass process is typically required for the welding of a gap between two plates. The limitations placed upon this process due to the narrow gap
and the inaccessibility of the melt pool puts special constraints on the monitoring of the process. In most cases the region of interest (ROI) can only be captured through the use of smaller High Dynamic Range (HDR) cameras. In order to develop and test both the traditional and DNN machine vision methods in this research, a suitable amount of data is required that captures the TIG arc welding process across a varied degree of performance or failure modes. This was achieved through the development of a experimental setup with TWI Ltd following on from previous work [3] [4] over a 4 phase trial into defects in a narrow gap TIG arc welding process. The experimental setup can be seen in figure 1. The narrow gap torch is to the left of the image in between the two plates to be welded. The camera that is used for the collection of video data is the XVC-1000 HDR camera which is specialised for this use-case. The camera has a range of +140dB, the frame rate used in this study was 30 fps and it has a resolution of 1024x1280. The camera is attached to an arm from the welding tool, which allows it to keep a steady frame on the ROI. This experimental setup was then used as a platform for the creation of a number of process defects that can occur during TIG welding activities. These include, excessive sidewall fusion (ESWF), lack of sidewall fusion (LSWF), and splatter defects used throughout the remainder of this research. The study was undertaken in 4 phases. The initial phase consisted of parameter testing to find a set that induced a certain type of defect, 5 parameter sets were established in this part of the trial. The main parameters in TIG welding are Background Current (A), Primary current (A), Voltage (V), Travel speed (mm/min) and Wire feed speed (mm/min), all of these except the voltage were varied. Phase 2 constitutes the testing of these parameter sets and getting a larger collection of defects, these included ESWF, LSWF, splatter, travel speed excessive and current excessive. Phase 3 was the final preliminary testing before the main trials in Phase 4 which is the most informative and well documented phase and the one of focus in this project. The Phase 4 dataset consisted of 150 welding videos, each video depicting a full pass of the weld purposefully inducing a certain type of defect. Upon sorting of the original 150 video dataset, 25 sample videos of a normal welding classified as ‘Good’, along with 9 for LSWF, 17 for ESWF and 6 for Splatter were extracted and used to validate and evaluate the traditional and DNN methods developed. The following section outlines the motivation and methodology used to develop each method.

III. NARROW GAP TIG ARC WELD PROCESS MONITORING

A. Traditional method

Traditional methods of defect classification are often based around high level feature extraction from images of the weld pool, in a sense it is used to gain information in a similar way to how a manual welder may diagnose a defect visually. These methods incorporate many of the elements of machine vision. Typically described as feature engineering, the goal is to provide the best state for feature extraction, before using these features with either machine learning methods or engineered tools tailored to the specific fault you wish to identify or process to track. Though we wish to track or detect our failures modes outlined previously (ESWF, LSWF, and splatter), we first must capture a key component of TIG welding, the arc tip. A common approach, and one currently implemented in industry, is the concept of seam tracking, which tracks the arc tip in relation to the centre or sidewall [5], as this determines how much material is fused to the sidewall. This is the cause of both LSWF and ESWF, from being too far and too close to the wall respectively, thus the first step is to employ object tracking on the weld tip.

1) Arc tip tracking: In this use-case, it is best to prioritise accuracy and robustness in a tracker, as there is generally minimal movement from frame to frame, since there is only minimal movement in the arc tip. Therefore, the Multiple Instance Learning (MIL) [6] tracker fits these criteria and was the most likely to perform well in the welding setting. It works by looking in the neighbourhood around the object defined in the previous frame and identifying multiple potential positive examples. This means that it is much less likely to lose track of the object while in a noisy environment. Though the tip tracking method has been established, it doesn’t provide useful information about the defect alone, as the tip location has been transformed to pixel space of the supplied image / video frame. In order to detect LSWF and ESWF, the relative tip position with respect to the sidewalls must be calculated and thus the sidewalls must be defined.

2) Sidewall detection: The main feature of the sidewalls is that they are straight, they remain within the frame in all of the defect examples within the dataset. The Hough Transform, developed in the 1960s, is a machine vision technique which detects the presence of simple shapes within an image like straight lines or circles. The algorithm is a three-step process: the first is the initialisation of an accumulator, which is simply an array to store the lines detected, the second is to apply a canny filter which defines the edges detected in the image using a pixel intensity threshold set in the parameters of the Hough Line function. Thirdly, as the pixels are transformed to Hough space, the points with more intersecting sinusoids than
a specified threshold are defined as a line. This technique has been used in the welding context to aid in the seam tracking as seen in [6] and [7].

The optimal values for the resolution were found to be $\theta = 0.75$ and $\phi = \pi/60$ for the Hough algorithm. Additionally, a threshold value for the canny filter must be set. Due to the amount of noise in the videos, this had to be fine-tuned by testing a tight range of values. This parameter had the largest effect in the accuracy and number of the lines and was determined to be best at 35. To filter out any superfluous lines it was assumed that any horizontal lines or any past a certain angle can be ignored. Using these parameters the image on the right of figure 2 was obtained overlaying the detected lines in green, it shows a strong line detection on the far left and far right of the image. The collection of lines defining the sidewall is analysed and the most appropriate line is selected, which is simply done by using the closest line to the centre of the image in both cases.

3) ESWF and LSWF detection: Having the capabilities to identify both the arc tip position and the sidewalls of the narrow gap structure we can then move on to a more direct detection of specific failure modes. ESWF and LSWF can both be classified using the relative arc tip position, in order to determine the state of the weld, data must be collected to analyse the position of the tip during the occurrence of each defect. As the welding technique consists of passes on the left and right sidewall, this means that ESWF can take place on both sides so a sample from each was included. The distance to each side is calculated by finding the x coordinate intercept of a horizontal line passing through the tracked arc tip and the defined sidewall, in relative terms the left sidewall is taken as 0. In order to the define arc tip position defect detection zones, a sample set of videos was used to collect the tip position data for each defect (ESWF / LSWF) type and the data used to determine zones or interest.

These zones are shown in figure 3 overlayed on a good weld frame. The defined good weld zones make up 20-25% of the gap in green, these are either side of the LSWF zone which is around 9.5% depicted in blue. ESWF is a much larger classification zone being 60% bounding the good weld zone and extending to the boundary of the sidewall highlighted in red, though the arc tip is unlikely to be within a certain distance of the plate due to physical constraints.

4) Splatter detection: Splatter is the last of the defect types, there have been many different attempts to detect splatter in a frame from the welding process, however, a novel method of splatter detection is proposed. This method relies on the characteristic that, in most cases, splatter takes the form of a circular ball of material that has been ejected from the weld pool.

The method chosen is the Hough Gradient corresponding to [8] and the parameters include $dp$ which determines the accumulator array resolution (the larger the value of $dp$ the smaller the array); $minDist$ determining the minimum distance between two detected circles, if this depends on how close the splatter is expected to be. However, as the welding process should be terminated at the first detection of splatter, this parameter shouldn’t matter. Parameter 1 is the threshold value used for edge detection like in the line transform; parameter...
2 defines the number of circles detected; the minimum and maximum radius of the circles to be detected must also be set which was defined by manually measuring the size of splatter examples. These parameters required a large amount of tuning through trial and error testing, due to the noisy nature of the videos.

5) Pre-processing and integration: The outlined techniques integrate to form a final tool that can detect and classify the four classes. The process starts with the supply of frames from a video or real-time source which are run through a number of pre-processing steps to clear up the images and determine the ROI. Next, the first feature that is extracted is the tip position using the MIL tracker, an estimation of the sidewalls is made for the calculation of the relative tip position. This is a separate function called from within the tool that returns a right and left sidewall estimation, this must be converted from polar coordinates for plotting. Now that the sidewall and tip location have been extracted the classification of defects can take place, this starts with the splatter detection function which deploys the Hough Circle transform to detect instances of circles in the frame. Next the relative tip positions and established defect classification zones are used to detect ESWF and LSWF, this is also a function called from the main loop. If either is detected their respective error code is set, once this classification process is complete the sidewalls, tracker, circles, error code and other identifying information is overlayed on the original frame for feedback to the operator. An example of both ESWF and splatter detection can be seen in figure 4.

B. DNN method

The past section described the development of traditional machine vision methods for the process monitoring and defect detection in narrow gap TIG arc welding. However, recent advancements in DNN machine vision methods has meant we can look to remove the need for complex feature engineering and learn directly from the data. This section outlines this approach, through the development of a DNN based model for the same classification of errors, but through a supervised learning approach.

In this instance the Residual Network architecture with 50 layers or ResNet-50 [9] is used and is shown in figure 5. There are a number of approaches which can be used when applying DNN models to a machine vision challenge such as this. The choice in this paper is to exploit the use of pre-trained models and simply retrain only the final layers of the model. This allows us to use the rich features of the pre-trained models but for a new use case, in this instance defect classification. The typical 1000 neuron layer is replaced with a 4 neuron layer, one for each class of interest (ESWF, LSWF, splatter, good).

1) Dataset generation: In order to train the DNN model effectively, we need to create a dataset of images and labels for it to be trained on. This dataset was created from the original 150 welding videos of phase 4 trials. A training set of 16,000 images was used, which represented an even number of examples for each class (ESWF, LSWF, splatter, good). To effectively train and evaluate the model, a split between training and testing data; of 75% to 25% training and testing respectively is used, as it provides a good balance in the two sets. This meant that 4,000 frames needed to be produced for each type of class, and 1,000 of these are to be used as validation data.

2) DNN model training: After the dataset had been prepared for training, the training parameters must be decided. First the number of epochs that the training will run over: this is essentially how many times the training algorithm with pass the data through the model, and each pass should see an increase in accuracy until it converges. The number of epochs must be high enough to ensure the model has enough time to train, however, the more epochs the longer the model takes to train. Here, 5 epochs are deemed sufficient to fully train the network in a reasonable amount of time for convergence. The model is trained using PyTorch on an RTX 2060 Super 8GB, which kept training times to a minimum.

IV. RESULTS AND DISCUSSION

Traditional machine vision methods and more recent state-of-the-art DNN methods are equally capable of being applied
to monitor joining processes as discussed earlier. The next step is to investigate how both these developed methods can be applied and perform a comparative assessment of their performance. The first step is to create a new dataset of images built from phase 4 sample videos of each of the classes (ESWF, LSWF, splatter, good) not used previously. This consisted roughly of 1,200 frames or 40s of video footage for each class. The resulting set of images were then combined into a single video and used as a real-time source for evaluating each of the developed methods. The goal for each method being to correctly predict the specific class of failure that each frame has been labelled with. The final video also contained a mix of splatter and ESWF labels, as discussed previously ESWF is often the cause of splatter defects. The results are plotted over the sequential frames in figure 6 for both traditional and DNN methods, where the actual class label is shown in red, whilst the prediction is in blue. The accuracy and precision statistics can also be seen in table 1 and give a good comparison with respect to each defect highlighting some interesting insights into the comparative testing.

On analysis of the classification figures there are some clear similarities in the two solutions, on the whole both have a large majority of correct classifications which are represented by the red line. The first obvious similarity is the misclassification of the good weld class frames, in both solutions there are a significant amount of LSWF and ESWF frames that are misclassified as good. This is less so with the misclassification of ESWF, the first 1200 frames, as class good using the DNN with just 5 cases, however, their presence is still meaningful. As explained prior ESWF and splatter cause issues with the classification in both cases, in comparison the traditional method has a 5.3% higher accuracy with 97.8%. The traditional method however has a much lower precision due to the large amount of splatter frames that are misclassified. However, this doesn’t happen the other way round when classifying splatter defects, having a precision of 1.0 in the traditional case due to the minimum frame detection threshold.

The DNN model classifies the splatter much more accurately than in the traditional case, with an 87% accuracy compared to the 82.3% roughly 5%. The second video in the sample was the good weld class examples, in this case both classifiers achieved an extremely high accuracy, 100% for the DNN and 99% for traditional, meaning that false positives of defects very rarely occur. LSWF was another class where the traditional methods outperformed the DNN, achieving a 6.3% higher result though the precision was much lower. Overall in terms of the performance in classification the DNN outperforms the traditional methods with 92.1% compared to 82.3% this is roughly a 10% improvement. In addition to this the DNN achieves a higher precision and accuracy in the good classification though there is only 1% in the accuracy statistic. However, the traditional result is significantly reduced by the 37.2% accuracy of splatter, it is therefore important to return to the observation about how the splatter was created during the TWI trials, splatter was a side-effect of ESWF. Taking this into account if it is assumed both classification results are correct the accuracy statistic for the splatter class changes...
considerably. The splatter classification accuracy in this case will now go to 100% as the only other classification was ESWF. The final accuracy calculations then are 95.4% for the DNN and 98% for the traditional methods, the traditional method would then be the most accurate in this case. In order to fully address this issue the use of multi-class classification is needed which allows for multiple different defects to be detected in one frame. As the models are being assessed for a real-time application the processing speed can be compared, this needs to be sufficiently fast to keep up with the process. The traditional solution ran at 15fps average, and is limited by the tracker deployed, if needed a different tracker could be used to keep up with the frame rate of the camera. The DNN was much faster, processing at 140 fps, this was with access to sufficient GPU processing power, and it is more than enough for a real-time use-case. In order to fully compare these solutions, it is important to also look at some other metrics.

Many factors should be taken into account when assessing the quality of the solutions presented, their viability for industry is a big comparing factor in the serious look at industrial application. Most importantly the robustness of the models as out of the experimental setting there are many other factors at play and potential sources of noise. The DNN is the clear favourite in this respect due to the ability to retrain the network and generalising ability, with traditional methods very specific pre-processing methods were deployed that were tested on the data available meaning these would most likely need rethinking if a different setup is used, as they are much more susceptible to noise. The DNN however at first was overfitting the data but if it was implemented in industry the longer it is in use, the more available data becomes and with further training, hopefully better generalisation and accuracy. In terms of cost both solutions would run on any computer that can integrate the welding camera, this means there is little need for any other specialised equipment which keeps costs low. The lifecycle of a TIG welding machine is very long and therefore the solution must be robust enough to deal with minimal changes in the environment and require little adjustments. Overall the DNN is more viable for use in industry though for more specific simpler tasks with minimal budget the traditional solution may be the best option. The traditional method is based on manual feature detection and therefore involves the application of prior knowledge about the defects being extracted. The feature extraction and pre-processing methods in this solution took a significant amount of time, whereas in the DNN method the pre-processing is minimal and there is no need for feature extraction as the model undertakes this in an all-in-one solution this this task. Due to this fact the DNN is seen as a blackbox solution meaning it is very difficult to tweak individual parameters in the network if something needs to be changed and a whole retraining is required. To conclude, understandably both approaches have their merits, and each has been demonstratively shown to work with a high degree of accuracy. Going forward it may be possible to explore integrating other sensing modalities beyond just vision and further investigate the ease in which this can be achieved through traditional or DNN methods.

V. CONCLUSION

In conclusion, two solutions have been proposed for defect monitoring and classification on a narrow gap TIG welding dataset. The first solution was based on traditional machine vision methods, and the second applied an DNN based solution. These solutions were tested, analysed and compared in order to evaluate their classification performance and potential for industrial implementation assessing the strengths and weaknesses in each case. Overall, there is no obvious choice when it comes to which method to implement as there are clear benefits and drawbacks to both methods as their processes of classification are so different. Both methods achieved high accuracy across the 3 defects we wished to track. The main benefit of the DNN solution is its adaptability and ease of setup, whereas in the case of traditional the solution is much more efficient. This provides evidence that there are a range of defect classification techniques able to achieve high accuracy and reliability in real-time analysis and characterisation of manufacturing processes, for which there is a great need.

ACKNOWLEDGMENT

The authors acknowledge the support of the Royal Academy of Engineering (RAEng) and Airbus under the Research Chairs and Senior Research Fellowships scheme (RCSRF1718/5/41). The authors also acknowledge the funding support from UKRI as part of the EPSRC Future Electrical Machines Manufacturing Hub.

For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

REFERENCES


![Table I](image)


Exploitation of a Multiband Weighted Envelope Spectrum for Railway Axle Bearing Fault Diagnosis

Bingyan Chen¹,*, Zewen Zhou², Weihua Zhang¹, Fengshou Gu¹,², Adam Bevan³, Philip Shackleton³

¹ State Key Laboratory of Traction Power, Southwest Jiaotong University, Chengdu 610031, China
² Centre for Efficiency and Performance Engineering, University of Huddersfield, Huddersfield HD1 3DH, UK
³ Institute of Railway Research, University of Huddersfield, Huddersfield HD1 3DH, UK

* Corresponding author (E-mail: chenbingyan@my.swjtu.edu.cn)

Abstract—Accurate diagnosis of axle bearing faults is a key part of ensuring the operating safety of railway rolling stock. The methodology combining spectral coherence and envelope analysis proved to be efficient for detecting bearing defects. However, the fault-related information of the axle bearing is disturbed by the complex noise in the train and exhibits a multi-band distribution, which cannot be revealed through the traditional envelope spectrum based on spectral coherence. To overcome these limitations, a spectral coherence-based multiband weighted envelope spectrum (MWES) is presented for the detection of axle bearing defects, enabling efficient extraction and integration of information in multiple frequency bands. Firstly, a signal-to-noise ratio defined in the frequency domain (FDSNR) is introduced for evaluating the richness of fault-related information along the spectral frequency in the spectral coherence. Secondly, a weighting function is designed for the FDSNR using a threshold denoising strategy that enhances the fault-related information while removing the interfering components. Finally, MWES for axle bearing diagnosis is constructed by weighted integration of the normalized spectral coherence along spectral frequency. The fault diagnosis performance of MWES is validated by experimental data from axle bearings. Comparisons with state-of-the-art envelope spectrum techniques highlight the advances of the developed MWES.

Keywords—railway axle bearings; envelope analysis; spectral coherence; multiband weighted envelope spectrum; fault diagnosis

I. INTRODUCTION

Axle bearing is one of the core components of railway rolling stock bogies to ensure that the wheelset runs along the rail, and its defect or failure has a serious influence on the safe and stable transportation of trains. Fault diagnosis is an effective means to discover potential risks in time, safeguard the stable performance of rolling stock and improve transportation efficiency. However, due to dynamic service conditions, the response signals related to bearing faults are often seriously polluted by complex vibration noises generated by track irregularities and vehicle body vibrations [1]. These unfavorable factors seriously hinder the diagnosis of railway axle bearing faults. Thus, it is very worthwhile to establish more robust axle bearing fault diagnosis techniques for complex interference noise.

The vibration response of a faulty bearing consists of periodic pulses induced by the fault, and the corresponding characteristic frequencies in the envelope spectrum can indicate the occurrence of the fault and its location [2]. Envelope spectrum analysis is a proven fault diagnosis method for bearings, gears and other rotating components. In this methodology, a resonant frequency band that is closely associated with the fault is first determined, the signal is then band-pass filtered and ultimately the envelope spectrum detection is employed on the filtered data to determine the fault. The selection of the filtering band has a significant effect on the diagnostic capability of the envelope spectrum. The fault features of rolling bearings have been widely recognized as impulsive and cyclostationary. The established envelope analysis methods can be divided into three categories: methods targeting impulsiveness [3]–[5], methods targeting cyclostationarity [6]–[8], and methods targeting both impulsiveness and cyclostationarity [9], [10]. In the approach targeting the impulsive nature of bearing fault signals, bands with high amplitude random pulses rather than repetitive transient pulses are usually selected. In contrast, the methods targeting the cyclostationarity exhibit high resistibility to Gaussian noise and random transient impulse when extracting bearing fault features. Nonetheless, these schemes are not true cyclostationary analysis techniques or methods.

Spectral correlation is a typical analytical tool for second-order cyclostationarity. Spectral coherence is obtained by normalizing the spectral correlation [11], [12]. Both SC and SCoh are capable of simultaneously disclosing the resonant frequency components and modulation frequency components induced by bearing defect in a dual-frequency representation. The one-dimensional spectrum produced by averaging SC over full spectral frequencies is shown to be equivalent to the squared envelope spectrum [11]. Furthermore, the enhanced envelope spectrum (EES) [13] generated by averaging SCoh over entire spectral frequencies was found to exhibit better performance than the traditional squared envelope spectrum in bearing diagnostics. However, EES is ineffective in reducing the contamination of interfering noise and exhibits unsatisfactory performance at low signal-to-noise ratios (SNRs). To solve this problem, the integral of SCoh over a narrow spectral frequency band associated with bearing faults was introduced to generate
the improved envelope spectrum (IES) [14] for fault detection and diagnosis, which enhances the ability to extract cyclostationary features. Unfortunately, IES can only obtain fault-related information in the optimal resonant band, and cannot effectively fuse fault feature information located in multiple resonant spectral frequency bands. Mauricio and Gryllias [15] proposed a combined IES (CIES), which can extract fault-related components located around multiple resonant spectral frequencies according to the distribution differences of fault information in the spectral frequency components of SCoh. However, the construction of CIES does not remove the spectral frequency bands dominated by fault-irrelevant components, which is not conducive to the elimination of interfering noise. Zhang et al. [16] developed a weighted envelope spectrum (WES) by applying a weighting function to SCoh and used it to detect faults of locomotive wheelset bearings. Although a threshold is introduced into the design of the weighting function to make the part zero, the threshold is a fixed value, making it difficult to effectively adapt to different operating conditions.

To solve the above problems of the SCoh-based envelope spectra and realize the cyclostationary feature extraction under complex working conditions, a performance-enhancing envelope spectrum approach constructed by SCoh, called multiband weighted envelope spectrum (MWES), is presented for the defect detection of railway axle bearings in this study. Firstly, a robust feature evaluation index is constructed to measure the fault-related information distribution along the spectral frequency axis of SCoh. Secondly, a threshold denoising strategy is applied to the obtained evaluation indicator to derive a weighting function. Finally, the SCoh is normalized and integrated in a weighted manner to construct MWES for fault diagnosis. MWES can effectively reduce interference noise components while extracting bearing fault features located in multiple, rather than one, optimal spectral frequency bands. The experiments of different railway axle bearings confirm the efficiency of MWES, and the comparative results with two existing methods demonstrate the advantages of MWES in axle bearing fault diagnosis.

II. THEORETICAL BACKGROUND

A. Spectral Coherence

SC is based on the instantaneous autocorrelation function. Suppose \( x(t) \) is a one-dimensional signal with second-order cyclostationarity, its instantaneous autocorrelation function is expressed as follows [13]:

\[
R(t, \tau) = E\{x(t)x(t-\tau)\}
\]

where \( E\{\} \) indicates the expectation operation, \( * \) denotes the complex conjugate, and \( \tau \) is the time delay. In theory, SC is the result of the two-dimensional Fourier transform of \( R(t, \tau) \), defined as follows [13]:

\[
S(\alpha, f) = \int R(t, \tau) e^{-j2\pi(\alpha \tau + f t)} dt d\tau
\]

where \( f \) and \( \alpha \) denotes the spectral frequency and cyclic frequency, respectively. The direct calculation of SC is very complicated and time-consuming, therefore, an estimation algorithm is usually used to calculate the SC of a signal. Currently, several estimation algorithms for SC have been proposed. To improve the computational efficiency, this study adopts a fast SC estimation technique using short-time Fourier transform [13].

To enhance the weak cyclostationary components, in practical applications, the magnitude of SC is usually normalized to be between 0 and 1 to generate SCoh, which is defined as follows [12]:

\[
\gamma(\alpha, f) = \frac{S(\alpha, f)}{\sqrt{S(0, f)S(0, f-\alpha)}}
\]

B. Generalized Integrated Spectrum

Since the two-dimensional spectrum is not convenient to implement fault diagnosis, the one-dimensional spectrum obtained by SCoh integration is often used. EES is an enhanced bearing fault diagnosis approach compared with the traditional squared envelope spectrum. However, an equal weight scheme is applied to the whole frequency components of SCoh and ignores differences in the richness of fault information within different spectral frequency components when constructing EES. Recently, Chen et al. [17] presented the generalized integrated spectrum (GIS) based on SCoh for bearing fault diagnosis, as follows:

\[
GIS(\alpha) = \int_0^\infty |\gamma(\alpha, f)| w(f) df
\]

where \( w(f) \) indicates a general weighting function that evaluates the distribution of defect features in the entire spectral frequency components. Obviously, the design of the weighting function is the key to constructing a robust envelope spectrum.

III. MULTIBAND WEIGHTED ENVELOPE SPECTRUM

The vibration measurement signal of railway axle bearing has the unique characteristics of complex composition and low SNR, which may lead to the failure of the aforementioned SCoh-based envelope spectrum methods. To achieve accurate diagnosis of axle bearing faults under complex disturbances, an MWES method is established. In the construction of MWES, both the distribution difference of fault information and the adaptive threshold denoising are considered to enhance the extraction of useful information and improve diagnostic ability. Based on these two useful information enhancement schemes, MWES can not only effectively extract fault-related cyclostationary features but also minimize the interference of fault-irrelevant components. The specific steps for implementing the presented MWES are described as follows.

Step 1: Acquire bearing vibration data

The vibration acceleration dataset of train axle bearings under constant rotation speed is collected by data acquisition instrument and accelerometer at a specific sampling rate.

Step 2: Calculate spectral coherence

First, an appropriate window length is set according to the length of the analyzed data, and the maximum cyclic
frequency to observe is determined based on the sampling frequency and the characteristic frequency of interest. Then, the two-dimensional SCoh of the acquired vibration data is calculated by using the fast estimation technique [13].

**Parameter selection for computing SCoh:** There is no specific criterion for how to set the window length, but mainly based on experience. To control calculation error and shorten calculation time, the length of the window used should be much shorter than the length of the signal to be analyzed under the condition of satisfying the required resolution of spectral frequency [13]. The maximum cyclic frequency to scrutinize corresponds to the upper limit of the cyclic frequency band that can be observed. In general, the upper limit of the cyclic frequency to be observed should be greater than three times the bearing characteristic frequency to be recognized for reliable diagnosis, i.e., at least three harmonic components of the cyclic frequency of interest are included. However, a too large maximum cyclic frequency consumes more computation time.

**Step 3:** Estimate bearing fault information distribution

A quantitative indicator with clear physical meaning, called frequency domain SNR (FDSNR) [18], is introduced to quantitatively estimate the fault information richness in the spectral frequency components of SCoh to obtain a one-dimensional function of spectral frequency. FDSNR is dedicated to evaluating the characteristic frequency of interest and its harmonic components in the envelope spectrum, defined as:

\[
\text{FDSNR}(f) = \frac{L-H}{H} \sum_{n=1}^{H} \left[ \frac{\max_{\alpha \in A_{n}} \left| \gamma(\alpha, f) \right|^{2}}{\sum_{\alpha \in A_{n}} \left| \gamma(\alpha, f) \right|^{2}} \right] - \frac{\sum_{\alpha \in A_{n}} \left| \gamma(\alpha, f) \right|^{2}}{N_{\alpha} \max_{\alpha \in A_{n}} \left| \gamma(\alpha, f) \right|^{2}}
\]

where \( L \) is the number of discrete cyclic frequencies; \( H \) is the highest order of the harmonics of the fault characteristic frequency \( f^*_\text{c} \); \( A_n \) indicates a narrow cyclic frequency band centered at frequency \( h \cdot f^*_m \), which includes the frequency component \( h \cdot f^*_m \) and 3 frequency components on both sides [8]. To eliminate errors caused by slight rotational speed fluctuations, the maximum amplitude within a narrow cyclic frequency range is estimated as the amplitude of characteristic frequency. In this article, \( H \) is specified as the number of the harmonics of the detected frequency \( f^*_m \) in the cyclic frequency band \([0, \alpha_c]\), and \( \alpha_c \) is assigned as the upper limit of the cyclic frequency to scrutinize.

**Step 4:** Construct weighting function

Due to the uneven distribution of fault information, the FDSNR value varies in different spectral frequency components of SCoh. The richer the fault information of the spectral frequency components of SCoh, the larger the corresponding FDSNR, and the smaller otherwise. To extract useful information and eliminate interference components simultaneously, a denoising threshold is designed to adaptively recognize spectral frequency components with different information richness, as follows:

\[\text{thres} = \text{mean}\{\text{FDSNR}(f)\}\]

where \( \text{mean}\{\} \) is the mean operator. Considering that the mean is a measure of the average level of the sequence, the mean of the estimated FDSNR is used as the threshold. The advantage of using the mean as the threshold is that it can be adaptively adjusted according to the measurement data in various working conditions.

Based on the achieved FDSNR function and adaptive information threshold, the weighting function is designed as follows:

\[
\phi(f) = \begin{cases} 
\text{FDSNR}(f), & \text{FDSNR}(f) \geq \text{thres} \\
0, & \text{FDSNR}(f) < \text{thres}
\end{cases}
\]

**Step 5:** Generate multiband weighted envelope spectrum

Finally, applying the designed weighting function to SCoh, the MWES is generated by integration to disclose the fault-related characteristic frequencies and diagnose the axle bearing faults, defined as follows:

\[
\text{MWES}(\alpha) = \int_{0}^{f_{\alpha/2}} \phi(f) \cdot |\gamma(\alpha, f)| df
\]

Similar to WES and CIES, MWES also evaluates the distribution differences of fault information along the spectral frequency axis and thus can capture bearing fault-associated features located in multiple spectral frequency bands. Differently, in the construction process of MWES, a self-adaptive information threshold that is immune to different operating conditions of railway axle bearings is introduced. Compared to the constant threshold used by WES, this scheme can effectively reduce interferences from the noise components that are not related to faults.

**IV. RAILWAY AXLE BEARING FAULT DIAGNOSIS**

To validate the actual performance of the presented MWES in bearing diagnostics, the experimental datasets (vibration acceleration) of actual railway axle bearings collected from a passenger car wheelset bearing test bench are analyzed.

**A. Experimental Setup**

The bearing test bench shown in Figure 1(a) mainly includes the foundation, a wheelset mounted with two axle bearings, driving device, loading device and a matching control system. A damaged bearing is fitted to one end of the train wheelset and a healthy bearing is fitted to the other end. The axle bearings are double-row tapered roller bearings, which are used in China’s high-speed railway electric multiple unit. Experiments were conducted on the axle bearings with damaged outer ring and roller respectively at constant rotational speed. The local defects with a width of 0.6 mm and a depth of 0.2 mm were artificially implanted on the surfaces of the bearing outer ring and roller. The number of rollers, pith diameter, roller diameter and contact angle of the faulty axle bearings are 17, 187.21 mm, 26.69 mm and 12.08°, respectively. The acceleration sensor for data acquisition was installed above
the axle box, as depicted in Figure 1(b). The sampling rate of bearing experimental data is 12800 Hz. The length of each experimental data is 8192 sampling points.

Figure 1. (a) Railway wheelset bearing test bench and (b) the installation location of the acceleration sensor.

B. Axle Bearing Outer Ring Defect

Firstly, a test at 883 r/min was conducted on a railway axle bearing with an outer ring defect. The fault characteristic frequency of the bearing outer ring $f_o$ is about 107.5 Hz. Figure 2(a) displays the measured experimental data of the test axle bearing with a defect implanted in the outer ring. The window length is specified as 64 data points and the upper limit of the cyclic frequency is designated as 600 Hz. Figure 2(b) depicts the two-dimensional SCoh of the collected acceleration signal, in which the bearing fault-related resonant frequency bands and fault-associated characteristic frequencies are difficult to detect. EES of the acquired bearing experimental data is shown in Figure 2(c), where the fault characteristic frequency of the outer ring $f_o$ and its harmonic frequencies (denoted in red dashed lines) are not prominent due to the disturbance of interfering noise.

MWES is performed on the measured bearing experimental data presented in Figure 2(a) to detect the outer ring defect. Figure 3(a) depicts the estimated FDSNR distribution along the spectral frequency axis. It shows that the feature information associated with seeded outer ring defect of axle-bearing is largely distributed around 0.4 kHz, 2.6 kHz and 4.7 kHz. MWES of the bearing experimental data is shown in Figure 2(c), in which $f_o$ and its first four harmonic frequencies $2f_o$, $3f_o$, $4f_o$ and $5f_o$ (denoted by the red dotted lines) are clearly identifiable. Such results show that FDSNR accurately estimates the fault-related information distribution and efficiently integrates information across multiple bands. MWES successfully confirmed a fault that was preset in axle bearing outer ring.

Figure 4 illustrates the results obtained by applying the recently developed weighted envelope spectrum (WES) method [16] to the acquired experimental data. Figure 4(a) exhibits the bearing fault information distribution estimated by the mean fault-symptomatic peaks to the mean noise (MFPN). In the WES method, a fixed value, i.e. MFPN=1, is adopted as the threshold for noise reduction. Although MFPN achieves a similar effect to FDSNR for estimating the distribution of axle-bearing fault information, the MFPN curve is above threshold 1, as shown in Figure 4(a). Because the fixed threshold fails to remove the defect-irrelevant frequency components, the $f_o$ and its harmonic frequency $4f_o$ (denoted by the red dotted lines) are still polluted in WES, as shown in Figure 4(b). These comparisons further verify the efficiency and advantage of the presented MWES technique in diagnosing the outer ring defect of railway axle bearings.

Figure 2. Results of experimental data of axle bearing with defective outer ring: (a) signal waveform, (b) SCoh, and (c) EES.

Figure 3. Results of MWES on the experimental data of outer ring damaged axle bearing: (a) FDSNR and (b) MWES.

Figure 4. Results of WES on the experimental data of outer race damaged axle bearing: (a) MFPN and (b) WES.
C. Axle Bearing Roller Defect

Furthermore, a railway axle bearing with a faulty rolling element was examined at a speed of 1165 r/min. The roller spin frequency (characteristic frequency of roller fault) \( f_b \) is about 66 Hz. Figure 5(a) shows the collected experimental data (vibration acceleration) of the bearing with a defective roller. The maximum cyclic frequency to scrutinize and window length are designated as 360 Hz and 64 data points, respectively. The two-dimensional SCoh of the collected bearing experimental data is illustrated in Figure 5(b), where the resonance spectral frequency bands and bearing defect-induced characteristic frequencies are hard to distinguish. Figure 5(c) depicts the EES of the bearing experimental signal. Although the roller spin frequency \( f_b \) and its harmonics can be recognized slightly, their spectral lines (denoted by the red dotted lines) are not prominent compared with the interference noise components.

The presented MWES approach is firstly used for the measured bearing vibration signal depicted in Figure 5(a). Figure 6 displays the achieved defect detection results. Figure 6(a) exhibits the estimated FDSNR values across the entire spectral frequency range, in which two obvious peaks can be detected around 2.8 kHz and 4.6 kHz, showing that the useful information associated with roller defect of axle bearing is mainly located around the peaks of these two spectral frequencies. In Figure 6(b), the fault characteristic frequency of the bearing roller \( f_b \) and its first four harmonic frequencies (represented by the red dotted lines) can be clearly recognized, meaning that the presented MWES scheme efficiently diagnoses the roller defect of the railway axle bearing. In addition, the sidebands on both sides of \( f_b \) and \( 2f_b \) can be detected in the spectrum, which is a typical symptom of bearing rolling element fault.

Furthermore, the WES approach is also imposed on the bearing experimental data depicted in Figure 5(a) for defect detection. Figure 7(a) displays the MFPN for estimating bearing fault information distribution. The estimation effect of MFPN is similar to that of FDSNR, but the fluctuation is relatively large and the entire MFPN curve is above the threshold \( \text{MFPN}=1 \). Although the roller spin frequency \( f_b \) and its several harmonic components could be identified from WES shown in Figure 7(b), their spectral lines (represented by the red dotted lines) are not as pronounced as those in MWES. The reason for this result is that the fixed threshold \( \text{MFPN}=1 \) adopted by WES fails to reduce the pollution of signal components unrelated to the roller defect. These results further illustrate the validity of FDSNR in evaluating the richness of bearing fault information and the advanced nature of MWES in diagnosing the rolling element defect of railway axle bearings compared with WES.

Figure 5. Results of the experimental data of roller damaged axle bearing: (a) signal waveform, (b) SCoh, and (c) EES.

Figure 6. Results of MWES on the experimental data of roller damaged axle bearing: (a) FDSNR and (b) MWES.

Figure 7. Results of WES on the experimental data of roller damaged axle bearing: (a) MFPN and (b) WES.

V. CONCLUSIONS

A novel SCoh-based multi-band feature extraction method, MWES, is presented for the diagnosis of axle bearings in this study. According to the distribution difference of fault information in different spectral frequency components, the weighted average is performed on the spectral frequency components of SCoh, which overcomes the limitations of traditional equal-weighted average and single frequency band selection methods. The FDSNR is introduced as a measure of fault-related information, and an adaptive threshold denoising strategy is designed to highlight bearing defect features by
removing the interference of defect-irrelevant components. The combined application of variable frequency band weight and adaptive threshold denoising enables MWES to maximize the reduction of contamination from interfering noise components while fully extracting bearing defect information in multiple spectral frequency bands. The validation results on the experimental datasets of different axle bearings show that: (1) MWES is capable of efficiently extracting axle bearing defect features from complex wheelset axle-box vibration signature and identifying axle bearing defects; (2) compared with EES and WES, MWES deliver high capability to disclose second-order cyclostationary features under complex interferences and diagnose different defects of railway axle bearings. Thus, MWES is a promising fault diagnosis method for axle bearings of railway rolling stock.

REFERENCES


Material Removal Rate Prediction with Phase Sensitive Variables Selection and Phase Partition

Chunpu Lv  
Department of Automation  
Tsinghua University  
Beijing, China  
lvcp16@mails.tsinghua.edu.cn

Tao Zhang  
Department of Automation  
Tsinghua University  
Beijing, China  
taozhang@tsinghua.edu.cn

Huangang Wang  
Department of Automation  
Tsinghua University  
Beijing, China  
hgwang@tsinghua.edu.cn

Abstract—In the Chemical Mechanical Planarization (CMP) process of the semiconductor industry, high accuracy prediction of Material Removal Rate (MRR) is essential to achieve wafer-to-wafer (W2W) control. Therefore, data-driven Virtual Metrology (VM) methods for MRR prediction have received much attention. Traditional methods focus on extracting features from raw data along the time dimension. However, a CMP process is a batch process with multi-phase and uneven-length characteristics, and different phases have different operating points and data correlations. In this paper, a novel VM method based on phase partition and feature selection is proposed. The phase-sensitive variables are selected by a wrapped approach and combined with a clustering algorithm for phase partition. After extracting the phase features, a suitable subset of features is selected using multiple feature selection methods and input to a regressor for prediction. The results in the PHM16 Dataset validate the effectiveness of the proposed method, which has higher prediction accuracy compared with the methods without phase partition.

Index Terms—Material Removal Rate, Phase-Sensitive Variables, Phase Partition, Feature Selection

I. INTRODUCTION

In the Chemical Mechanical Planarization (CMP) process of the semiconductor industry, an important variable, the Material Removal Rate (MRR), is difficult and costly to measure. Numerous researches have been devoted to establishing relationships between a key indicator and multiple process variables, and to estimating the indicator that is difficult to measure. This process is called Virtual Metrology (VM).

The VM of MRR mainly includes two types of methods, physics-based methods and data-based methods. Preston, Luo&Dornfeld proposed formulas for calculating MRR, respectively [1] [2]. However, it is difficult for the physics-based approaches to fully and accurately describe the relationships between MRR and all variables. Therefore, data-based methods are more popular among researchers. Cai, Feng et al. used the k-Nearest-Neighbour (kNN) combined with Gaussian Process Regression based models to fuse the reference sample information, and predict the final MRR by Multi-Task Gaussian Process (MTGP) [3]. They also proposed Just-in-time (JIT) model-based strategy, which uses Support Vector Regression (SVR) and Particle Filter (PF) to estimate and update mixing results [4]. Li et al. proposed a novel decision tree-based ensemble learning algorithm, integrating regression results from different regressors by stacking technique [5]. Jia et al. proposed an adaptive methodology based on the group method of data handling (GMDH) type polynomial neural networks [6]. Zhao et al. transformed the data into multidimensional information, then determined feature subsets by feature importance assessment, and finally used stacking fusion for prediction [7]. Feng et al proposed to track the slow drift of manufacturing processes using an online Bayesian Auto-Regression eXogenous model and the parameters are trained off-line and are updated online using Bayes’ rule [8].

CMP is a typical batch process that has two important characteristics, the multi-phase characteristic and the uneven-length duration. An operation batch contains several phases, and the relationships of variables are different between them. Some researchers have proposed phase partition methods based on Principal Component Analysis (PCA) [9] [10] [11]. These methods require that the duration of each batch is equal, which is difficult to achieve in actual processes. Luis A et al. proposed a multi-purpose partition clustering procedure, Wrapped K-Means (WKM) algorithm, with an imposing hard sequence constraint for clustering and phase partition [12]. Li, Zhao et al. proposed a sequential time slice alignment-based unequal-length phase identification and modeling method [13]. Luo et al. proposed a sequence-constrained fuzzy c-means clustering algorithm, which divides the trajectory data of phase-sensitive variables into several fuzzy operational phases [14]. Guo et al. proposed a Sequential Moving Principal Component Analysis method to explain the dynamic multi-phase characteristics of the sampled data [15].

From the above review, it can be found that the MRR prediction methods do not consider the contribution of phase features. Meanwhile, clustering-based phase partition methods depend on phase-sensitive variables, but there is a lack of research on this subject.

The main contributions of this paper are as follows. First, a wrapped phase-sensitive variables selection method based on model errors and variable numbers is proposed. Second, the WKM algorithm is combined with the Partition Performance on Combination Index (PPCI) to complete the phase partition for each batch, while avoiding too detailed partition. Third, the phase features are extracted and a sequential feature selection method based on the Akaike Information Criterion (AIC) is used to identify a suitable features subset to be input to
the Light Gradient Boosting Machine (LGBM) Regressor for prediction analysis. Compared with the existing literature, the phase-sensitive variable selection method was proposed for the first time, and the proposed method achieved higher prediction accuracy.

The rest of this paper is organized as follows. Section II describes the methods. Section III presents the experiments and results. Section IV is the conclusion.

II. METHODS
A. Wrapped Phase-Sensitive Variables Selection

The selection of phase-sensitive variables requires a combination of model errors and the number of variables. The phase-sensitive variables need to change significantly at the time point of phase switching so that more phase information can be obtained in feature extraction. Therefore, a more accurate prediction result can be obtained based on the phase partition of the phase-sensitive variables.

In this paper, we reference the idea of wrapped feature selection and combine the phase-sensitive variables selection with phase partition, feature extraction, and the prediction of MRR. In other words, the selection index of phase-sensitive variables is closely related to the prediction accuracy. In the CMP process, different process variables can be grouped according to their nature, including pressure-group, flow-group, usage-group, rotation-group, etc. Different groups of variables are selected to participate in the phase partition, feature extraction, and regression prediction, and the mean square error (MSE) of prediction is used as one of the considerations for phase-sensitive variables selection.

At the same time, the number of phase-sensitive variables is considered. If the number is too large, the complexity of the model is enhanced and the computational effort increases. The trajectories of the same group of variables are similar, and the trajectories of different groups of variables are more different. Therefore, the set of variables with the highest prediction accuracy will be selected as the phase-sensitive variables.

B. Phase Partition With WKM and PPCI

In improving the WKM algorithm, Luo et al. changed the K-means clustering to fuzzy c-means clustering and gave the membership degree matrix. In the fuzzy c-means clustering algorithm, the Sum of Squared Errors (SSE) is obtained by weighting the m-th power of the membership degree of each sampling point to each phase. This leads to a smaller contribution to the SSE from the transition phase sampling points with lower membership degrees. Transition phase sampling points are important for phase partition. Therefore, this method leads to blurred phase boundaries [14] [16]. Jiang et al. proposed that the combination of SSE and PPCI balanced the clustering error in phase partition with the number of phases. The use of PPCI avoids the over-partition of transition phases, accurately identifies stable phases, and lays the foundation for phase feature extraction [17].

Considering an unequal-length 3D data matrix \(X(I \times J_{PS} \times K_i)\), where \(I\) represents the number of batches, \(J_{PS}\) represents the number of phase-sensitive variables, and \(K_i\) is the sampling length of the \(i\)-th batch. The data of the \(i\)-th batch is \(X_i = [x_{i1}, x_{i2}, \ldots, x_{iK_i}], x_k \in \mathbb{R}^{J_{PS} \times 1}, k = 1, 2, \ldots, K_i\). Given the number of phases \(C\), the goal is to find the phase partition corresponding to the smallest SSE and determine the boundaries, center, and the number of sampling points for each phase. SSE is calculated as (1) and (2):

\[
SSE = \sum_{c=1}^{C} \sum_{k=b_c}^{b_c+n_c-1} (x_k - \mu_c)^T(x_k - \mu_c) \tag{1}
\]

\[
\mu_c = \frac{1}{n_c} \sum_{k=b_c}^{b_c+n_c-1} x_k \tag{2}
\]

where \(b_c, \mu_c, n_c\) represent the starting point, center, and the number of sampling points of the \(c\)-th phase, respectively.

The initial phases are determined using the average distribution of global error or the average distribution of the sampling length. Then the WKM algorithm is used to move sampling points of each phase. The sampling points are examined sequentially in sampling order, and the movement of sampling points can only occur between adjacent phases, thus ensuring that the clustering satisfies the time order constraint. The premise of moving is to decrease the SSE. When sampling point \(x\) is removed from phase \(X_i\) to phase \(X_{i+1}\), the change of SSE is calculated as shown in (3):

\[
\Delta SSE(x, c, w) = \frac{n_w}{n_w + 1}(x - \mu_w)^T(x - \mu_w) - \frac{n_c}{n_c - 1}(x - \mu_c)^T(x - \mu_c) \tag{3}
\]

The WKM algorithm can only divide a single batch under the given clustering number \(C\), while \(C\) cannot be determined. PPCI has been proposed to fill this gap. PPCI performs a weighted average of the normalized SSE and \(C\). Too large SSE or too large \(C\) will lead to an increase in PPCI. With PPCI, the appropriate number of phases is selected to avoid the detailed partition of transition phases. The WKM-PPCI algorithm focuses on the core of phase partition on stable phases.

C. Feature Selection based on Akaike Information Criterion

After completing the phase partition, the features of the phases are extracted, including the initial values, means, standard deviations, medians, areas under the curves, and peak-to-peak values of the process variables, as well as the duration of the phase and the starting time. These features together with the global features of the batch constitute the initial features set. However, the number of initial features is large and there are redundant or irrelevant features, so feature selection is needed.

Feature selection in this paper is divided into two steps. The first step is filtered feature selection, which filters out features with a large proportion of data vacancies, single values, low variance, high linear correlation, and low importance.

After that, the wrapped approaches provide a more fine feature selection. The wrapped feature selection continuously
selects a subset of features, trains the learner, and evaluates the subset based on the performance of the learner. In addition to the exhaustive method, feature search strategies in subsets include random search and sequence search, etc. Sequence search is currently more widely used, which mainly includes forward search, backward search, and forward-backward search. In the forward search strategy, the initial feature subset is empty, and one feature is added in each round until the set number of features is reached. In the backward search strategy, the initial feature subset contains all the features, and one feature is removed in each round until the set number of features is reached. The forward-backward search completes two steps simultaneously, adding features and removing features. In this paper, the Sequence Forward Floating Selection (SFFS) algorithm [18] is used to complete the search strategy.

The SFFS algorithm can select a given number of feature subsets, but the number needs to be judged by other methods. In this paper, a method is designed to determine the number of features based on the Akaike Information Criterion (AIC) [19]. AIC can average the complexity and fitting ability of the model, thus preventing overfitting. AIC is calculated as (4):

$$AIC = N \log(MSE) + 2k$$  \hspace{1cm} (4)

where $k$ denotes the number of features, $MSE$ denotes the mean square error, and $N$ represents the number of samples. Applying AIC to the SFFS algorithm, if the curve of AIC changes from decreasing to increasing and cannot continue to decrease to a new minimum value within a given number of cycles, the search is stopped and the feature subset corresponding to the minimum AIC value is taken. AIC-SFFS is described as Algorithm 1.

D. Prediction: Light GBM

Light Gradient Boosting Machine (LGBM) was proposed by Ke, Meng et al. as an improvement to the Gradient Boosting Decision Tree (GBDT) algorithm [20]. As the amount of data gets larger, GBDT scans all the data to estimate the information gained at possible split points, increasing the computational complexity. LGBM designs two new techniques to implement weighted sampling. One is Gradient-based One-Side Sampling (GOSS), using a downsampling approach to maintain the accuracy of information gains. Another one is Exclusive Feature Bundling (EFB). In practical applications, the feature space is usually very sparse, so mutually exclusive features (which rarely take nonzero values simultaneously) are bundled to reduce the number of features. Experiments show that LGBM has a large advantage in terms of computational speed and memory consumption.

III. EXPERIMENTS

A. Datasets

The PHM16 Dataset is derived from the 2016 Prognostics and Health Management (PHM) Data Challenge and provides a set of CMP process data. The dataset includes a training set, a test set, and a validation set, containing 1981, 424, and 424 wafer batches, respectively. Each batch contains 25 variables, 6 of which are device information variables and 19 are process variables. The process variables include 5 groups, usages, pressures, speeds, flows, and status. The batch length varies, with an approximate range of 300–400. The trajectories of different variables in a batch are shown in Fig. 1, and it is clear that the batch data can be divided into multiple phases. Therefore, the dataset suggests that the CMP process is an uneven-length multistage batch process.

PHM16 Dataset can be divided into 3 modes based on two information variables, the chambers, and the stage, and the division is shown in Table I. The methods in this chapter will take Mode I as an example.

![Fig. 1. Trajectories of features in a batch.](image-url)
B. Phase-Sensitive Variables Selection

The wrapped phase-sensitive variables selection requires a combination of phase partition, feature selection, and prediction accuracy calculation steps. Therefore, this operation is included in C-E below. Among the 19 process variables, the usage variables show a stepwise increase with time, a status variable is an irregular 0-1 variable, while the pressure variables, the rotation variables, and the flow variables all show certain multi-phase characteristics. In comparison, the phases of the pressure variables are clearer, and the phase features extracted after the phase partition based on the pressure variable provide higher prediction accuracy. Therefore, six pressure variables were used as phase-sensitive variables. The results of phase partition based on the pressure variables and the flow variables are shown in Fig. 2 and Fig. 3.

C. Phase Partition

In Mode I, the data located in chamber 4 are less missing and have strong phase stability, while the data located in chambers 5 and 6 have missing and very irregular phases, so the phase partition is only for chamber 4.

The WKM algorithm was used for phase partition. For each batch, the initial phase boundaries and phase lengths are determined using the TS algorithm, and the phase centers, as well as the SSE, are initialized according to (1) and (2). Next, along the sampling time, whether the sampling data in the first half of each phase need to be moved to the former phase and whether data in the second half needs to be moved to the latter phase are examined. The judgment is based on the fact that, if the SSE decreases after the move, i.e., \( \Delta L < 0 \) in (3), then the move is accepted. The phase boundaries, phase lengths, phase centers, and SSE are updated after the moves of the samples in all phases are completed.

In the phase partition algorithm, multiple phase numbers are used to obtain a set of SSEs, and the SSEs and phase numbers are weighted by PPCI to select the number of phases corresponding to the minimum PPCI. The relationship between PPCI and phase numbers is shown in Fig. 4.

It can be seen that the phases of a batch include stable phases with long duration and transition phases with short duration, which can be distinguished by the given minimum duration \( L_S \). The number of stable phases is counted as shown in Table II, and the plural of the stable phases is taken as the standard number of stable phases, which is 5 in the experiment. Then, the data of each batch is matched into 5 stable phases according to the distances of the phase centers.

D. Feature Extraction and Selection

In the experiment, different features are extracted from phases. Also, the same features are extracted for the whole
data of chamber 4, and the data of chambers 5 and 6, which do not complete the phase partition. There are 70 features for each phase or chamber. For the feature set of each phase or chamber, a filtered feature selection method is first used to remove single-valued, low variance, high linear correlation, and low importance features. The experimental results show that 25, 43, 43, 39, 21 features are retained for the 5 phases and 47, 34 features are retained for the 2 chambers, respectively.

The remaining features will be selected by AIC-SFFS. Feature selection starts from 1. The AIC is used as the performance evaluation index, and if the subset of the new features has enhanced performance, then the AIC and the optimal features subset are updated. Otherwise, the next quantity of feature selection is carried out. If the AIC is not updated after multiple rounds (given a maximum step size of 20), the loop is ended and the computation process ends. Fig. 5 show the results of AIC-SFFS.

![Sequential Forward Floating Selection](image)

(a) The mean squared error.

![AIC](image)

(b) The AIC.

Fig. 5. The AIC based sequential forward floating selection.

As the number of features increases, the prediction error gradually decreases, which is due to the fact that more features contain more information. However, when the number reaches a certain range, the rate of error decline gradually slows down and the space for model effectiveness enhancement gradually decreases. At this time, the AIC limits the number of features. Fig. 5(b) reveals that the curve of AIC first declines and then rises. In the declining stage, the change in prediction error plays a major role, and in the rising stage, the increase in the number of features takes up more weight. In this experiment, the final number of features is 27.

### E. Prediction Results Using Light GBM

The selected features are input into the LGBM regressor. The learning rate is set to 0.1, and the number of boosted trees is 100. The prediction results of LGBM and other regression algorithms, such as the Linear Regression, Support Vector Regression (SVR), Extra Trees, GBDT, and AdaBoost are shown in Table III. The prediction results show that the LGBM algorithm has higher prediction accuracy than others. In addition, for most of the algorithms, the inclusion of phase features helps to improve the prediction accuracy.

<table>
<thead>
<tr>
<th>Regressor</th>
<th>MSE(Phase)</th>
<th>MSE(NoPhase)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>6.22</td>
<td>6.17</td>
</tr>
<tr>
<td>SVR</td>
<td>10.45</td>
<td>7.81</td>
</tr>
<tr>
<td>Extra Trees</td>
<td>6.08</td>
<td>6.43</td>
</tr>
<tr>
<td>Random Forest</td>
<td>6.20</td>
<td>6.38</td>
</tr>
<tr>
<td>GBDT</td>
<td>6.38</td>
<td>5.89</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>6.83</td>
<td>7.31</td>
</tr>
<tr>
<td>LGBM</td>
<td><strong>5.54</strong></td>
<td><strong>6.05</strong></td>
</tr>
</tbody>
</table>

### TABLE III

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Chamber LGBM</th>
<th>Chamber FS LGBM</th>
<th>Phase LGBM</th>
<th>Phase FS LGBM</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>5.95</td>
<td>5.62</td>
<td>5.75</td>
<td><strong>5.54</strong></td>
</tr>
<tr>
<td>$R^2$</td>
<td>0.855</td>
<td>0.863</td>
<td>0.860</td>
<td><strong>0.865</strong></td>
</tr>
</tbody>
</table>

Table IV shows the regression results using different feature sets, and with or without performing the feature selection algorithm. “Chamber” indicates that the feature set contains only features extracted from two chambers. “Phase” indicates that the feature set contains features extracted from five phases. “FS” indicates that the AIC-SFFS was used for feature selection. Comparing the regression results, we can find that the inclusion of phase features is helpful to improve the prediction accuracy. Meanwhile, the LGBM algorithm can output the ranking of feature importance. Among the top 10 important features, half of them are extracted from the phases. Therefore, the phase partition provides more information for the prediction of MRR. Meanwhile, the prediction results with feature selection using AIC-SFFS are better than those without feature selection. Reasonable feature selection can remove redundant features and reduce the prediction error.

The comparisons of the proposed method with the existing methods are shown in Table V. The regression prediction results of the proposed method under the index of MSE are better than the existing methods. The experiments show that phase-sensitive variables selection, phase partition, and feature selection help to improve the MRR accuracy and reduce the prediction error. The prediction results of the proposed method selection are shown in Fig. 6, it can be seen that the proposed algorithm provides accurate predictions for MRR. For the prediction of MRR, the accuracy of the proposed
method is high and the error is small. Meanwhile, the proposed method develops phase partition, feature extraction, and feature selection around the original data, so it is more interpretable. However, compared with other methods, the proposed method has more steps and takes a longer time, especially the feature selection, which uses multiple rounds of selection and increases the computational complexity.

<table>
<thead>
<tr>
<th>Methods</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrated Model with GMDH [6]</td>
<td>5.89</td>
</tr>
<tr>
<td>GMDH [6]</td>
<td>5.90</td>
</tr>
<tr>
<td>Online Bayesian ARX [8]</td>
<td>8.89</td>
</tr>
<tr>
<td>Proposed Method</td>
<td>5.54</td>
</tr>
</tbody>
</table>

Fig. 6. The prediction performance of the predicted method.

IV. CONCLUSION

In this paper, a novel prediction method for MRR in the CMP process is proposed. For the multi-phase and uneven-length characteristics of the CMP process, phase-sensitive features are selected, and phase partition is performed by the WKM algorithm combined with PPCI to extract more information from the phases. After extracting the phase features, the AIC-based wrapped feature selection method is used to reduce the complexity of the model. Finally, regression prediction is performed using the Light GBM algorithm. The experimental results on PHM16 Dataset validate the effectiveness of the proposed method. Future work will focus on the use of different clustering algorithms in phase partitioning and the prediction of MRR based on deep learning models.

REFERENCES

Rapid State of Health Estimation of Lithium-ion Batteries based on An Active Acoustic Emission Sensing Method

Zuolu Wang1, Kaibo Lu2, Xun Chen3, Dong Zhen4, Fengshou Gu1,* and Andrew D. Ball1

1Centre for Efficiency and Performance Engineering, University of Huddersfield, Huddersfield, HD1 3DH, UK.
2College of Mechanical Engineering, Taiyuan University of Technology, Shanxi 030024, PR China.
3General Engineering Research Institute, Liverpool John Moores University, Liverpool L3 3AF, UK.
4Tianjin Key Laboratory of Power Transmission and Safety Technology for New Energy Vehicles, School of Mechanical Engineering, Hebei University of Technology, Tianjin, 300401, P. R. China.
*Corresponding author: E-mail address: f.gu@hud.ac.uk

Abstract—Lithium-ion batteries have widely used as the power sources of electric vehicles (EVs). Accurate and rapid state of health (SOH) estimation in the battery management system (BMS) plays an essential part in improving the reliability and safety of electric systems. This paper develops an active acoustic emission (AE) sensing technology for non-intrusive and rapid battery SOH estimation. The proposed method takes consideration into the changing internal battery material properties under different levels of degradation. In this method, the power ultrasound is used to propagate into the layered battery and excite different AE events of the battery under different cycles. The AE transducer from the opposite side of the battery can actively sense the elastic waves that reflect the life status. This allows more state information to be captured in a wide frequency band for effective SOH estimation. The results indicate that the RMS of the AE signal can be indicative of battery SOH, and the frequency band 270-300 kHz can provide a more linear SOH estimation under various discharging stages. It is validated that the developed technique can achieve rapid and reliable SOH estimation of lithium-ion batteries.

Keywords- Lithium-ion battery; state of health; material properties; active acoustic emission; power ultrasound

I. INTRODUCTION

In order to alleviate the enormous pressure brought by the increasing consumption of fossil energy and propose solutions to climate change, energy storage systems and electric vehicles (EVs) have been extensively developed [1]. Lithium-ion batteries have developed into dominant energy storage units thanks to their higher power efficiency and lower self-discharge rate [2]. The battery state of health (SOH) can deliver the percentage of the current maximum available capacity to its nominal capacity, which indicates the aging level of the battery [3]. Accurate SOH estimation can result in a reliable battery management system (BMS) for improving the safety, battery performance and even extending the battery lifespan. Thus, developing an accurate and efficient SOH estimation method is still challenging and promising work for practical applications.

Currently, the battery’s SOH estimation has attracted extensive attention. In [4], the authors have given a comprehensive review on the battery SOH estimation approaches that can be generally grouped into three categories, including model-based, data-driven, and advanced sensing-based methods.

Model-based SOH estimation is implemented through modelling the battery parameters. The electrical equivalent circuit model (EECM) and electrochemical model (EM) are two popular models for accessing the battery SOH. In EECM, some essential variables, including polarization resistance/capacitance and internal ohmic resistance, vary with cycling and they are commonly used to characterize the battery SOH. Many filtering algorithms, such as EKF [5], PF [6], and AEKF [7], have been proposed for the identification of electrical parameters and the accurate SOH estimation. The EM considers the actual electrochemical reactions inside the battery, such as the solid-phase/liquid-phase diffusion process [8]. For example, Pseudo-Two-Dimensional (P2D) model and single particle model (SPM) [9] are two typical electrochemical models for presenting the dynamics of the battery, in which the active material concentration can be an effective factor to characterize the battery SOH. From different perspectives, both EM and EECM can duplicate the battery’s characteristics, but the large computational equations undoubtedly increase the complexity of state estimation.

The implementation of the data-driven approach is like a black box system that does not need to consider the complex electrochemical characteristics, and it only focuses on the relationship between the measured battery variables (e.g., voltage, current, and temperature) and degraded capacity for SOH prediction [10]. Due to the powerful nonlinear modelling, a variety of machine learning methods, such as the hybrid neural network that combines gate recurrent unit and convolutional neural network (GRU-CNN) [11], variable length input long-short term memory (VLR-LSTM) network [12] and hybrid kernel function relevance vector machine (HKRV) [13], have been employed to SOH evaluation of the battery. However, the data-driven method has high requirements on both computing power and large battery degradation datasets for various working conditions. This undoubtedly brings a huge challenge to the BMS due to its limited storage space and computing power.

Recently, some advanced sensing-based technologies have been developed and prove effective for more
convenient battery SOH evaluation. For example, electrochemical impedance spectroscopy (EIS) has been demonstrated effective to monitor the internal impedance under different cycles through the high frequency AC current excitation [14]. However, the acquisition of the impedance spectrum across a broad range of frequency excitations can increase the computational costs. Moreover, monitoring ultrasound technology is performed for SOH detection by focusing on the changes of battery material properties. Paper [15] proposed the ultrasonic pulse for the battery SOH estimation for the first time. It indicates that two parameters, including time of flight (TOF) and signal amplitude (SA) of the pulse echo, are highly correlated with battery SOH. Based on the variation of SA and TOF in the ultrasonic pulses, many studies have dedicated to SOH estimation [16], [17]. It is expected that the monitoring methods that take accounts into the battery material properties under different cycles can achieve easy, rapid, and accurate SOH estimation. However, the current widely used ultrasound-based methods are based on the single frequency pulse. Consequently, only a small amount of information can be considered for battery SOH evaluation.

In view of the changes of the battery material properties, this paper proposes an active acoustic emission (AE) sensing technology to achieve the rapid and cost-effective lithium-ion battery SOH estimation for the first time. The developed rapid detection method involves two main modules, including power ultrasound and AE. In particular, the proper power ultrasound is propagated to the layered battery and hence trigger elastic waves under different degradations. A wide AE transducer from the opposite side of the battery is applied to actively track the released elastic pulses that can characterize the battery life status. The experimental results indicate that the RMS of the AE signal can be easily calculated for the effective SOH estimation. The frequency band 270-300 kHz, around the 7th harmonics of the excitation frequency, can effectively provide battery SOH estimation at various discharging stages.

The rest of this paper is shown as follows: Section 2 introduces the proposed monitoring method in detail. Section 3 presents the experimental setup and detailed test procedures. The SOH estimation results from the experimental studies are presented in Section 4. Conclusions are presented in Section 5.

II. TEST THEORY

A. Multi-layered Structure

The pouch lithium-ion battery mainly consists of cathode, anode, and electrolyte [18], [19]. Thus, it is a multi-layered structure. The anode and cathode are coated on copper foils and aluminum, and they are divided by the separator as illustrated in Fig. 1.

![Multilayered structure of the pouch battery.](image)

Various layered materials have different properties and hence cause a variety of acoustic impedances. Different degrees of transmissions and reflections are induced when sound waves pass through two different materials. The reflection coefficient is expressed as [20]:

\[
R = \frac{Z_2 - Z_1}{Z_2 + Z_1} \quad (1)
\]

\[
Z = \rho c \quad (2)
\]

where \(Z_1\) and \(Z_2\) denote the acoustic impedance of two different battery materials. \(c\) denotes the wave velocity and \(\rho\) represents the density. The reflection coefficient is between 0 and 1, while \(1 - R\) stands for the transmission coefficient.

Battery material properties change as the battery degrades. Long-term cycling will consume the active material, which can result in the formation of solid electrolyte interphase and the loss of lithium ions. Experimental results have revealed that the elastic modulus and density of the electrode material decreased with the number of cycles [21]. Therefore, these irreversible changes in material properties can alter the reflection coefficient \(R\), which can influence the propagation of sound waves and introduce different levels of attenuation of sound transmission in batteries.

B. Proposed Active AE Sensing Method

The battery SOH can be treated as the function of the battery material properties. The proposed monitoring method combines the applications of AE and power ultrasound, which is to sense the changes in the material properties and then characterize the battery SOH. Compared to the ultrasonic sensor, the AE sensor can achieve broadband monitoring and hence collect more adequate information for SOH estimation of the lithium-ion battery.

The appropriate power ultrasound is applied to excite the battery and trigger different elastic waves under different degradations. Its role is included in mechanical and chemical effects on the battery. For example, the mechanical effects, such as shear force and shock wave, can act on the materials inside the battery and trigger different responses. Moreover, mechanical effects can accelerate the movement of lithium ions and thus the chemical reaction. As a result, the proper power ultrasound that passes through the battery with different battery material properties can lead to micro deformations and excite different elastic pulses.
Figure 2. Acoustic emission spectrum with fundamental excitation frequency $f_0$.

Furthermore, the AE transducer installed on the opposite side of the battery is employed to actively track the released elastic pulses that can characterize the battery life status. Power ultrasound can introduce different AE events as battery material properties change. Thus, the AE sensor can capture the high-frequency waves and achieve the battery SOH estimation. Compared to the traditional vibration or sound sensor, the AE monitoring can effectively acquire the transient waves released by local materials in a high sampling frequency. In addition, more information can be acquired by the AE sensor in a wide frequency band [22]. As shown in Fig. 2, the harmonics of the fundamental frequency and the broadband signal can be obtained if the battery is acted by the excitation source with the fundamental frequency $f_0$.

III. EXPERIMENTS

A. Test Rig

To validate the performance of the developed technology for rapid battery SOH estimation, the power ultrasound and AE test rig were developed. Fig. 3 displays the schematic diagram of the designed test bench. It is composed of AE acquisition system, preamplifier, AE sensor, battery tester, temperature chamber, power ultrasound generator, vibrator, current clamp, data acquisition system, pouch battery, thermocouple, and host computer. The EBC-A20 battery tester satisfies the basic battery test, such as CC discharge, constant power discharge, CC-CV charge, and multi-step cycle test. The electrical parameters, such as terminal voltage, loading current, and surface temperature, are measured by the 16-bit data acquisition system with sampling frequency of 10 Hz. Fig. 3 (b) displays the layout of the targeted pouch battery in the chamber with constant temperature of 25℃. The thermocouple is adopted to measure the temperature of the battery surface. In the test, the vibrator excited by the ultrasonic generator is installed on the battery surface to excite the battery, in which the excitation frequency of the ultrasonic generator is 40 kHz. The response signal is collected by the AE transducer that can achieve the effective measurement at a wide range of frequency band between 60 kHz-400 kHz. Therefore, the sampling frequency and a digital band-pass filter of the acquisition system are set as 2 MHz and 20 kHz-500 kHz, respectively. The gain of the preamplifier is 40 dB. The experimental pouch battery is presented in Fig. 4 and the specifications are listed in Table 1.

B. Test Procedures

In the test, both the charging and discharging of the battery are carried out at 0.5 C rate. The cycle test was performed to degrade the battery. The proposed active AE sensing technology was carried out at the interval of 20 cycles. In the test, the battery is excited by the power ultrasound at three different points during the discharging stage, and the AE signals were collected accordingly. Table 2 describes the three points for the operations of ultrasonic excitation and AE signal collection. For each of the operations, the power ultrasound was first input to the battery and lasted for 10 s, and then the AE signal was acquired at the 5th second and the collection lasted for 5 s.

---

**Figure 3.** Schematic diagram of (a) power ultrasound and AE test rig, and (b) the layout of the pouch battery in the temperature chamber.

**Figure 4.** The experimental pouch battery.

**Table 1.** The Li-ion pouch battery specifications.

<table>
<thead>
<tr>
<th>Item</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>Polymer lithium-ion battery</td>
</tr>
<tr>
<td>Nominal capacity</td>
<td>2 Ah</td>
</tr>
<tr>
<td>Nominal voltage</td>
<td>3.7 V</td>
</tr>
<tr>
<td>Charge/discharge cut-off voltage</td>
<td>4.2 V/3 V</td>
</tr>
<tr>
<td>Size</td>
<td>7x43.3x62 mm</td>
</tr>
</tbody>
</table>

In the test, the battery is excited by the power ultrasound at three different points during the discharging stage, and the AE signals were collected accordingly. Table 2 describes the three points for the operations of ultrasonic excitation and AE signal collection. For each of the operations, the power ultrasound was first input to the battery and lasted for 10 s, and then the AE signal was acquired at the 5th second and the collection lasted for 5 s.
### TABLE 2. DESCRIPTION OF TEST TIME POINTS.

<table>
<thead>
<tr>
<th>Test Points</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point 1</td>
<td>Before discharging: SOC = 100%</td>
</tr>
<tr>
<td>Point 2</td>
<td>In discharging: SOC = 50%</td>
</tr>
<tr>
<td>Point 3</td>
<td>After discharging: SOC = 0%</td>
</tr>
</tbody>
</table>

Note: State of charge (SOC).

### IV. RESULTS AND DISCUSSIONS

#### A. Time and Frequency Domain Analysis

The battery was degraded to 100 cycles in this test. Fig. 5 illustrates the actual available battery capacity over cycles and the calculated SOH under different cycles. It can be seen that the battery SOH drops to about 96.87% after 100 cycles.

![Figure 5.](image)

#### B. SOH Estimation

Fig. 7 shows the calculated RMS at three measured points using various frequency bands under different SOHs. It is clear that the different frequency bands present different responses to the battery SOH. As mentioned in Section 2, the battery material properties change with ageing. For example, the density and elastic modulus of the electrode material reduce with the increase of degradation during which many lithium ions are consumed gradually. This can cause the attenuation of sound waves when propagating across the battery. Therefore, the calculated RMS under most of the frequency bands presents a downward trend with the degradation of the battery. As seen in Fig. 7 (a) (b) (c) and (g), the RMS calculated at the three points using the frequency bands, such as 70-400 kHz, 75-90 kHz, 110-150 kHz, and 270-300 kHz, decrease with the decreasing of SOH. However, the calculated results at some measured points using the frequency bands, including 150-180 kHz, 185-215 kHz, and 220-260 kHz, cannot effectively characterize the battery SOH. For example, the RMS of the measured AE signals in 150-180 kHz fluctuates largely and cannot give a linear relationship with battery SOH, which hence cannot offer a reliable battery SOH evaluation.

In contrast, the RMS calculated in 270-300 kHz exhibit a better linear correspondence with SOH at any testing points, as shown in Fig. 7 (g). In particular, Fig. 8 gives the calculated RMS at three testing points in the frequency band 270-300 kHz. It is noted that the RMS can characterize the variation of battery SOH relatively linearly. Therefore, the experimental results validate the feasibility of the designed monitoring method for the fast and convenient SOH evaluation of the lithium-ion battery.
This paper developed an active AE sensing technology by focusing on the varying battery material properties with degradations and it can effectively achieve the fast detection of the battery SOH. In this method, the appropriate power ultrasound aims to excite the battery and trigger different AE events under different levels of degradation, and the wide AE sensor can capture more state information in a wide range of frequency band. It is important to mention that the RMS can be an effective indicator for battery SOH estimation. Generally, the calculated RMS under various frequency bands exhibits a decreasing trend with the decrease of SOH. The experiment results reveal that the frequency band 270-300 kHz with $f_0$ can provide accurate SOH evaluation at three measured points. Therefore, it is validated that the developed monitoring technology can effectively achieve the fast detection of battery SOH.

REFERENCES


A Hybrid 1DCNN-SVM Model for Bearing Fault Recognition and Classification

Guoliang Wang, Hua-Liang Wei*
Department of Automatic Control and Systems Engineering
University of Sheffield
Sheffield, United Kingdom
w.hualiang@sheffield.ac.uk (*corresponding author)

Abstract—Rolling bearings play a significant role in the operation of rotating machinery. Finding bearing faults in the early stage can not only keep machinery running safely but also avoid economic loss. Traditional machine learning (ML) and deep learning (DL) are two important classes of fault diagnosis methods. The process of feature extraction is significant for ML based methods and many signal processing techniques have been used in this process. However, when dealing with massive and complex signals, features extracted by these techniques may not fully describe the characteristics of faults. Convolutional neural network (CNN) models are applied in many areas, including image identification and classification, pattern recognition, and speech processing. In this paper, a novel hybrid model by combining one dimensional CNN (1DCNN) and support vector machine (SVM) is proposed for bearing fault diagnosis and classification, which is called 1DCNN-SVM model or method. The first part of the model, 1DCNN, is used to extract features of raw data, whereas the second part, SVM, is employed to carry out the fault recognition and classification tasks. The evaluation of the proposed method is operated based on a real dataset. The experimental results show that the fault diagnosis accuracy of proposed method based on the hybrid model is highly improved compared to the traditional SVM based methods.

Keywords—fault detection; fault diagnosis; fault detection; machine learning; deep learning; 1D CNN

1. INTRODUCTION

Bearings are the key components of many rotating machineries and any failures of bearings may result in economic losses even production accidents. Therefore, the demand for fault detection and diagnosis of rolling bearings is ever-increasing. Many classical machine learning (ML) methods, for example, support vector machines (SVMs), decision trees, k-Nearest Neighbors (kNN), artificial neural networks, and DL models, are widely used in fault detection and diagnosis. Recently, deep learning (DL) models, for example, convolutional neural networks (CNNs), deep belief neural networks (DBNs), recurrent neural networks (RNNs), and deep adversarial and transfer learning, have been employed for bearing fault diagnosis [1].

Classical ML-based fault diagnosis methods can be used to build a relationship between the collected data and the machine health status using the extracted features from the raw data. Therefore, feature extraction is a key step in classical ML-based fault detection and diagnosis approaches. Features can usually be extracted from time-domain, frequency-domain, and/or time-frequency domain [2]. SVM as one of the ML models was initially designed for binary classification problems, but its later variants can also be applied to solve multiclass classification problems. SVMs have been widely used for fault diagnosis. In [3], a method based on SVM and principal component analysis (PCA) was employed for the diagnosis of wind turbine, where PCA was used to extract features, and a grid search (GS) algorithm was considered for the optimization of the model. The results for simulation data showed that this method could achieve very high precision when solving fault diagnosis problems. In [4], a shuffled frog leaping algorithm (SFLA) was used for the parameter tuning, then the SFLA-SVM was compared with the PSO-SVM method. The results showed that the SFLA-SVM model had a better performance than the traditional PSO-SVM model. In [5], a novel optimization algorithm called sparrow search algorithm (SSA) was used for parameter tuning and the SVM model optimized by SSA was compared with many other SVM models. The experimental results show that the accuracy of fault diagnosis based on the SSA-SVM model is significantly increased.

CNN, as one of the DL frameworks, has been developed and applied in many areas, for example, image identification and classification, speech processing, and pattern recognition, because of the good performance in feature extraction [6]. CNN was initially designed to process 2D images but it can also be used to process 1D signals. Many fault diagnosis methods based on CNN [7]-[10] have been proposed. Except for CNN model, other DL models, e.g., stacked auto-encoder [11][12], deep adversarial [13], and long short-term memory [14] are used in fault diagnosis. However, unlike classical ML methods whose results can be interpreted or explained in one way or another, DL models usually cannot be explained. This may make DL-based fault diagnosis methods inapplicable to many fault diagnosis problems.

Classical ML-based models have good interpretability but the feature extraction process requires high expert knowledge and manual interference, whereas DL-based methods have powerful feature extraction capability but lack interpretability. To overcome the drawbacks of DL and classical ML-based methods, and take advantage of the two types of methods, this paper proposes a novel method for bearing fault diagnosis and classification based on a hybrid model combining 1DCNN and SVM. The main contributions of this paper include:
1) Automatic feature extraction for SVM model using 1DCNN.

2) Significant improvement of the accuracy of basic SVM model for handling multi-classification problems.

3) Increase of the interpretability of the outputs of the fully connected layer of 1DCNN models.

The remaining of the paper is organized as follows. A brief literature review on closely related methods is presented in Section II. Section III presents the proposed method. In section IV, experiments on real data are performed to evaluate the feasibility of proposed method and the hybrid model. In section V, brief concluding remarks are given.

II. A BRIEF LITERATURE REVIEW ON RELEVANT METHODS

A. Support Vector Machine

SVM [15] was proposed to solve classification problems based on the structural risk minimization principle and the key is to find a hyperplane that can make the margin between two different datasets as large as possible. If the hyperplane is defined as \( f(x) = \omega^T x + b \), and the data points in the input space are described as \((x_1, y_1), ..., (x_l, y_l)\) and the labels \( y_i \in \{-1, 1\} \), then the margin can be described by the following equation,

\[
\text{margin} = \frac{2}{\|\omega\|}
\]

where \(\|\omega\|\) is the root of the \(L_2\)-norm of \(\omega\).

Thus, the solution to this optimization problem can be formulated as:

\[
\min_{\omega, b} \frac{1}{2} \|\omega\|^2 \\
\text{s.t. } y_i(\omega^T x_i + b) \geq 1, \quad \forall i = 1, 2, ..., l
\]

By applying the method of Lagrange multipliers to (2), and letting the Lagrange multipliers be denoted by \(\alpha_i\), the solution for this optimization problem is rearranged as:

\[
\max_{\alpha} \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (x_i^T x_j), \\
\text{s.t. } \sum_{i=1}^{l} y_i \alpha_i = 0, \quad \alpha_i \geq 0 \quad \forall i = 1, 2, ..., l
\]

In real applications of SVM, a mapping function \(\phi\) is usually used to describe the input data points in a high-dimensional feature space. In addition, a relaxation variable \(\xi_i (\xi_i > 0)\) and the punishment factor \(C\) will be introduced to formula (2), such that the optimization problem becomes

\[
\min \phi(\omega) = \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{l} \xi_i \\
\text{s.t. } y_i(\omega^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i = 1, 2, ..., l
\]

Then, the optimization problem can be converted into a quadratic form:

\[
\min L(\alpha) = \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j K(x_i, x_j) - \sum_{i=1}^{l} \alpha_i \\
\text{s.t. } \sum_{i=1}^{l} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C
\]

where, \(K(x_i, y_i) = \phi(x_i) \cdot \phi(y_i)\) is the kernel function. Noted that the decision function of this optimization problem is given by:

\[
f(x) = sgn \left( \sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b \right)
\]

In real applications, the selection of kernel function can significantly impact the performance of SVM. In this paper, radial basis kernel function (Gaussian kernel) is used because it shows good general properties, e.g., powerful nonlinear mapping ability, fast convergence speed, and the involvement of fewer parameters [5]. The expression of the radial basis kernel function is:

\[
K(x_i, x_j) = \exp \left( -\frac{\|x_i - x_j\|^2}{2g^2} \right)
\]

where \(g\) is the scaling parameter of the basis function.

B. Convolutional Neural Networks

CNNs can not only be used to process 2D data (e.g., images) but also sequential data [16], and have shown powerful feature extraction ability. The structure of a basic CNN model comprises several convolution layers and pooling layers, a fully connected layer, and an output layer. A convolutional layer performs convolution operations by its kernels and these kernels have two significant characteristics: sparse connection and shared weights [6]; in the same convolutional layer, there are usually multiple convolutional kernels, and these kernels only act with the local region of the previous input with the constant weights. Let a d-dimensional input data can be denoted by \(x \in \mathbb{R}^d\); then on the \(j\)-th feature map, the \(i\)-th convolution result can be expressed by the following formula:

\[
c^i = x * o^i + b^i
\]

where the symbol \(*\) represents the convolution operation; \(o\) and \(b\) are the filters and basis separately. The following process is the non-linear operation, in which the activation function plays a key role. There are mainly three commonly used activation functions, namely, sigmoid function, tanh function, and rectified linear unit (ReLU), which are defined as follows respectively:

\[
sigmoid: f(x) = \frac{1}{1 + e^{-x}}
\]

\[
tanh: f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}
\]

\[
ReLU: f(x) = \max(0, x)
\]
Generally, the selection of activation function depends on the situation, but ReLU is widely used due to its faster computation speed and the inhibiting ability of gradient vanishing problem, so it is used in this paper.

In the pooling layer, the dimension of feature maps will be reduced and the number of parameters will also become smaller. Meanwhile, during the pooling operation, the main characteristics of feature maps will be maintained. After the convolutional layers and pooling layers, the extracted features go to the fully connected layer where the extracted features will be processed, then sent to the output layer for classification or other tasks.

III. PROPOSED METHODS

A. Proposed Method

In this section, the proposed fault diagnosis method is introduced. The framework of the proposed method consists of two parts: 1DCNN for feature extraction, then, SVM for fault recognition and classification. As shown in Fig. 1, the original signals are sampled and divided into training data and test data. The training datasets are sent into the pre-trained 1DCNN model directly without any signal processing steps. The feature vectors obtained from fully connected layer are used to train SVM for fault recognition and classification. The data splitting procedure is repeated until good feature vectors are obtained. Finally, the model is estimated based on the test feature vectors.

The pre-trained 1DCNN model consists of three convolution layers, three pooling layers, and one fully connected layer. A total of five trials were performed, and the 1DCNN model with the best performance is used for feature extraction. To prove the good feature extraction capability of 1DCNN, the graphical illustration of the output of each layer, generated by the t-Distributed Stochastic Neighbor Embedding (t-SNE) [17] method, is presented.

In Fig. 2, ten types of data, which are distributed in the space disorderly, are displayed. They are overlapped and it is hard to distinguish. Then, they are sent to Conv1, Conv2 and Conv3 for feature extraction and the distribution is shown in Figs. 3-5. After the convolution and pooling operation, the disordered data points are gathering. Finally, in the fully connected layer, 10 types of data can be distinguished clearly, as shown in Fig. 6. Therefore, it is feasible to use 1DCNN as feature extraction method.

<table>
<thead>
<tr>
<th>Layer type</th>
<th>Filter size</th>
<th>Number of filters</th>
<th>Strides</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>16</td>
<td>128</td>
<td>1</td>
</tr>
<tr>
<td>Pool1</td>
<td>2</td>
<td>128</td>
<td>2</td>
</tr>
<tr>
<td>Conv2</td>
<td>8</td>
<td>64</td>
<td>1</td>
</tr>
<tr>
<td>Pool2</td>
<td>2</td>
<td>64</td>
<td>2</td>
</tr>
<tr>
<td>Conv3</td>
<td>4</td>
<td>32</td>
<td>1</td>
</tr>
<tr>
<td>Pool3</td>
<td>2</td>
<td>32</td>
<td>2</td>
</tr>
<tr>
<td>Full</td>
<td>10</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

| TABLE I. PARAMETERS OF 1DCNN MODEL |

The feature extraction of raw data can be carried out by 1DCNN model automatically so there is no need to apply specific algorithms to select or extract features from the original signals. Original signals are sampled then they are divided into three parts, training data, validation data, and test data, with the ratio of 8:1:1. Training data are used to train the 1DCNN model, then the one with the best performance is selected based on the validation data. After that, test data are used to evaluate the selected model. The parameters of 1DCNN model are shown in Table I, where the convolutional, pooling, and fully connected layers are abbreviated as ‘Conv’, ‘Pool’ and ‘Full’, respectively.
C. Fault Recognition and Classification Based on SVM

After the process of 1DCNN, feature vectors of the original signals are obtained in the fully connected layer. Then, these vectors are used to train SVM for fault recognition and classification. From Fig. 1, feature vectors extracted from the training data are used to train SVM and there is no additional steps for optimization. In other words, punishment factor $C$ in (4) and parameter of kernel function $g$ in (7) are set to the default value, 1 and 0.1 respectively. There are many optimization algorithms that can improve the accuracy of SVM. In [18], GSA is used to improve the performance of SVM. Other optimization algorithms, for example, sparrow search algorithm [5] and particle swarm optimization [19], have also been used for SVM parameter optimization.

As will be shown in the experiments section below, the proposed 1DCNN-SVM can reach a very high accuracy without using any of these algorithms mentioned above to optimize the SVM parameters $C$ and $g$; it is shown that the default values $C = 1$ and $g = 0.1$ work very well for our proposed model.

IV. EXPERIMENTS AND RESULTS

A. Data Description

The Case Western Reserve University dataset [20] is considered in our experiments. The sampling frequency is 12 kHz and only the drive end accelerometer data are used. With different fault positions, we consider three fault types which are in ball, inner race, and outer race, respectively. Besides, there is one normal condition. All the data are recorded under four different work conditions, namely, 0 HP, 1HP, 2HP, and 3HP. In addition, for the fault data, there are three different fault types in diameters, that is, 0.53mm, 0.36mm, and 0.18mm. In total, there are 10 different types of data and the description of data can be seen in Table II. To have better experiment results, 10-fold cross-validation was used. For each health case, 800 samples are chosen in a random manner, and each sample has 1024 points. To realize 10-fold cross-validation, each health condition data is divided into 10 parts. In every trial, 9 parts are used as the training data and 1 part is used for test.

<table>
<thead>
<tr>
<th>Health condition type</th>
<th>Label classes</th>
<th>Number of samples (training/test)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>1</td>
<td>720/80</td>
</tr>
<tr>
<td>BF-0.18</td>
<td>2</td>
<td>720/80</td>
</tr>
<tr>
<td>BF-0.36</td>
<td>3</td>
<td>720/80</td>
</tr>
<tr>
<td>BF-0.54</td>
<td>4</td>
<td>720/80</td>
</tr>
<tr>
<td>IF-0.18</td>
<td>5</td>
<td>720/80</td>
</tr>
<tr>
<td>IF-0.36</td>
<td>6</td>
<td>720/80</td>
</tr>
<tr>
<td>IF-0.54</td>
<td>7</td>
<td>720/80</td>
</tr>
<tr>
<td>OF-0.18</td>
<td>8</td>
<td>720/80</td>
</tr>
<tr>
<td>OF-0.36</td>
<td>9</td>
<td>720/80</td>
</tr>
<tr>
<td>OF-0.54</td>
<td>10</td>
<td>720/80</td>
</tr>
</tbody>
</table>
B. Comparative Analysis

To construct SVM models for comparative experiments, the raw data must be processed by some signal processing methods to extract the features. Wavelet packet decomposition (WPD) is one of the commonly used feature extraction methods [21], it can decompose the raw data and provide the details and characteristics of raw data. In this paper, the original signals are decomposed at layer 3 so 8 feature vectors are obtained. After that, the feature dimension is reduced by performing principal component analysis (PCA) [22]. The percentage of each component that describes the raw data is shown in Table III. The criterion is to keep more than 95% of data information. From the table, it is obvious that component 1 to component 3 can describe more than 95% information. As a result, the 8-dimension feature vector is reduced to a 3-dimension feature vector. The model training process is shown in Fig. 7.

![Flow chart of the SVM model](image)

The punishment factor $C$ in (4) and the kernel function parameter $g$ in (7) have a great influence on the performance of the SVM model. To illustrate this, experiments under the following two conditions were carried out using SVM: 1) using the default values $C = 1$ and $g = 0.1$; and 2) using the optimized values suggested by the GS algorithm. The accuracies under the two conditions are shown in Table IV. Although the GS-SVM model can improve the accuracy of the conventional SVM model, running the model is very time consuming due to the optimization procedures. More importantly, the optimized GS-SVM model still has a large gap in performance compared with the proposed 1DCNN-SVM model.

C. Discussions on the Experiment Results

Good performances of GS-SVM have been reported in the literature (see e.g. [3]), unfortunately, the model does not show the expected high accuracy. This may be explained as follows. Firstly, the extracted features by WPD and PCA cannot describe the characteristics of the original signal very well. This is also the shortcoming of some fault diagnosis methods based on other traditional ML models. When dealing with massive and high-dimension data, it is hard to select suitable features to describe the raw data. Secondly, the SVM model was initially invented for binary classification problems. In this study, there are 10 types of data (faults) so SVM may not be able to achieve a high accuracy for this problem. To increase the accuracy, a multi-classification structure may be considered, for example, one against all and one against one [23].

The accuracy of 1DCNN-SVM model is higher than traditional SVM and GS-SVM even though the parameters $C$ and $g$ are set to their default values. It confirms the powerful feature extraction ability of 1DCNN. It has been shown that 1DCNN model can be used for bearings fault diagnosis [9][10], but the 1DCNN model does not have the explainability possessed by the SVM model. In CNN, it is difficult to know which characteristics have been extracted and how they interacted with each other to produce a prediction of the output. A softmax layer is usually used to calculate the probability of each class, but probability only provides a measure of the likelihood of each class but cannot give a clear physical meaning or explanation in real practice. By combing 1D CNN and SVM, the classification process is operated by SVM, which can be well explained, and each step of the implementation process has a clear physical meaning. This increases the interpretability of 1D CNN.

V. Conclusion

A novel hybrid 1DCNN-SVM model was proposed for bearing fault diagnosis. The experimental results indicate...
that the proposed approach significantly obviously outperforms the traditional SVM based fault diagnosis methods. However, the limitations of SVM may influence the performance of 1DCNN-SVM for multi-classification tasks. Therefore, other ML models, for example, DT and k-NN may be considered. In addition, RNN is also good at handling sequence data. So, in future investigations will be carried out to further explore the capability of 1DCNN based models for intelligent fault detection and diagnosis.

ACKNOWLEDGMENT

The authors acknowledge that this work was supported in part by NERC (Ref. NE/V001787/1), NERC (Ref. NE/V002511/1) EPSRC (Ref. EP/H00453X/1), and EPSRC (Ref. EP/I011056/1).

REFERENCES


Non-Singular Fixed-Time Sliding Mode Control for Unknown-Dynamics Manipulators Interacting with Environment

Haiyi Kong, Guang Li, Chenguang Yang

1 School of Engineering and Materials Science, Queen Mary University of London, Mile End Road, London E1 4NS, UK.
2 Bristol Robotics Laboratory, University of the West of England, Bristol, BS16 1QY, UK.

Abstract—In this paper, a robust fixed-time controller is designed for manipulators with unknown dynamics while interacting with environment. To realize compliance of the manipulator to the environment, an admittance model is adopted in the system. In the controller design, a non-singular sliding mode torque is introduced to achieve fixed-time convergence of the system state, so that the system has better tracking performance and faster response to external interaction than the system using traditional TSM controller. Meanwhile, RBFNN is employed to approximate the parameters in the dynamics. At last, simulation studies are conducted to verify the effectiveness of the proposed method.

Index Terms—Admittance control, non-singular sliding mode control, fixed-time convergence, neural network.

I. INTRODUCTION

Manipulators have received great attention in many fields due to their high degree of flexibility, such as industry, transportation, and medicine [1]- [2]. With the increasing complexity of application scenarios, higher requirements are put forward for manipulators, among which the ability to deal with the external environment is important.

Compliance is a way to interact with environment for manipulators, which can greatly improve manipulators' safety. During working process, if the manipulator is suddenly blocked by an external force, compliance can drive the manipulator to move along external force instead of generating larger torque against the environment [3]. When the external force is withdrawn, the manipulator will resume tracking the original trajectory. To realize the relationship between the environmental interaction position and interaction force, Mason proposed the admittance model in 1981 [4]. In [5]- [6], the adaptive admittance control is adopted to realize compliant behavior for redundant manipulators. In [7]- [8], an admittance model is employed in the system for cooperation between multi-manipulators.

When the manipulator is in contact with the environment, it will inevitably introduce undesirable interference torques, such as frictional torque and external disturbance. As for coping with external disturbance and system parameter perturbations, sliding mode control is a powerful method, which can effectively improve the robustness of the system [9]. In the early stage, linear hyperplane was used in the sliding mode control, where the system state can realize asymptotic stability [10]. Because the convergence time is infinite in theory, which is not realistic in the actual control. Hence, in this context, terminal sliding mode (TSM) was developed for achieving finite-time convergence [11]. However, traditional TSM controllers all suffer from the singularity problem. To tackle this problem, [12] proposed to switch the sliding mode between TSM and linear hyperplane, and [13] specified a non-singular open region. On the other hand, the convergence time of the finite-time TSM control depends on the initial states of the closed-loop system, which cannot be obtained beforehand. This drawback can be overcome by the recently upgraded TSM, that is fixed-time TSM control, where the settling time is uniformly bounded by a fixed time and independent of initial system states [14].

For general manipulator systems, computed torque control is usually used to achieve desired tracking performance [15]. However, the system's performance largely depends on the fidelity of the system model. In practice, accurate models of manipulators are difficult to obtain. In this context, much attention has been drawn to neural network, which can be used to approximate inaccurate or unknown dynamics of manipulators [16]. Radial basis function neural network (RBFNN) is one of the commonly used neural networks [17]. Because of its simple structure and strong generalization ability, RBFNN is especially suitable for systems requiring high real-time performance.

In this paper, we design a robust non-singular fixed-time controller for manipulators interacting with environment, where an admittance model is introduced to realize compliant motion. In the control design, a non-singular fixed-time switching manifold is employed to achieve desired fixed-time convergence for the system state, which results in better tracking performance of the system and faster response to external interaction than the system using traditional TSM controller. Meanwhile, RBFNN is employed to approximate
the unknown parameters in the dynamics. Simulation results are given to prove the effectiveness of the proposed method.

II. PRELIMINARIES

A. System Dynamics

The dynamic model of a \( N \)-joint manipulator can be expressed as

\[
\ddot{q} + C(q, \dot{q}) + G(q) + d = \tau + J^T F_e
\]

(1)

where \( \ddot{q}, \dot{q}, q \in \mathbb{R}^N \) represent acceleration, velocity and angle of the manipulator, respectively. \( M_q \in \mathbb{R}^{N \times N} \) stands for the inertia matrix in symmetric positive definite. \( G_q \in \mathbb{R}^N \) represents the vector of gravity. \( \tau \in \mathbb{R}^N \) is the joint torque representing the control input of the system. The interaction force between the environment and the end-effector of the manipulator is expressed as \( F_e \in \mathbb{R}^N \), which can be acquired by force sensor. \( J^T \in \mathbb{R}^{N \times N} \) is the transpose of the Jacobian matrix corresponding to the mapping from joint velocity to workspace velocity. \( d \) represents the external disturbance, which is assumed to be bounded \( \|d\| \leq d_m \). \( C_q \in \mathbb{R}^{N \times N} \) denotes the Coriolis matrix, which is chosen to satisfy

\[
\ddot{q} + C_q(q, \dot{q}) + G_q(q, \dot{q}) = 0
\]

(2)

The forward kinematic model of the manipulation is

\[
x = \Xi(q)
\]

(3)

where \( x \in \mathbb{R}^N \) is the pose of end-effector in Cartesian space with dimension \( N_c \). Taking the first and second derivative of (3) yields

\[
\dot{x} = J(q) \dot{q}
\]

(4)

\[
\ddot{x} = \dot{J}(q) \dot{q} + J(q) \ddot{q}
\]

(5)

B. Admittance Control

To achieve compliant motion of the manipulator when its end-effector follows the reference trajectory and interacts constantly with the environment, the reference position should be regulated accordingly. Admittance control is proposed for this purpose, that is, to maintain desired force contact by adjusting position. Commonly, the applied admittance model in the Cartesian space is chosen as

\[
D_A(\dot{x}_r - \dot{x}_d) + C_A(\dot{x}_r - \dot{x}_d) + G_A(x_r - x_d) = -F_e
\]

(6)

where \( x_r \) is the desired position in the original trajectory, \( x_d \) is the reference position produced by admittance model. (6) is the typical mass-damping-stiffness model, in which \( D_A, C_A \) and \( G_A \) are positive diagonal matrices. Based on (4) and (5), the admittance model (6) can be transformed into joint space

\[
D_A \dot{J}(q)(\dot{q}_r - \dot{q}_d) + C_A \dot{J}(q) + G_A \dot{J}(q) = -J^{-T}(q) \tau_e
\]

(7)

where \( \tau_e \) is the joint torque produced by the interaction with environment and \( \tau_e = J^T(q) F_e \).

C. RBF Neural Network

RBFNN has superior generalization ability that can approximate arbitrary linear or nonlinear function. The activation function commonly uses the Gaussian function:

\[
\mathcal{Y}_j(x) = \exp(|x - \sigma_j|^2 / 2\gamma_j^2), \quad j = 1, 2, \ldots, m
\]

(8)

where \( m \) denotes the corresponding number of neuron, \( \mathcal{Y}_j \) is the output of hidden layer, and \( \mathcal{Y} = [\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m]^T \in \mathbb{R}^m \). \( x \in \Omega_x \subset \mathbb{R}^n \) is the \( n \)-dimension input of the neural network. \( g_j \in \mathbb{R} \) and \( \sigma_j \in \mathbb{R} \) represent the width and center of Gaussian function of the \( j \)-th hidden node, respectively. \( \| \cdot \| \) is the norm of Euclidian distance.

For arbitrary continuous vector function \( f(x) \in \mathbb{R}^n \) approximating by RBFNN, it can be written in the form of

\[
f_i(x) = W_i^T \mathcal{Y}(x) + \epsilon_i(x), \quad i = 1, 2, \ldots, n
\]

(9)

where \( \epsilon_i \) stands for the small approximation error and satisfies \( \epsilon_i \leq \bar{\epsilon} \). \( \bar{\epsilon} \) is the known upper bound. We define \( \epsilon_i = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]^T \) and \( \mathcal{Y}^* = [\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_m]^T \in \mathbb{R}^{m \times n} \) where \( \mathcal{Y}^* \) represents the ideal weight of neural network.

Take \( \mathcal{W} \) as the estimation of \( \mathcal{Y}^* \). Correspondingly, we can denote the estimation of the continuous function as

\[
\hat{f}_i(x) = \tilde{W}_i^T \mathcal{Y}(x), \quad i = 1, 2, \ldots, n
\]

(10)

where \( \tilde{f}_i(x) = [\tilde{f}_1(x), \tilde{f}_2(x), \ldots, \tilde{f}_n(x)]^T \in \mathbb{R}^n \) and \( \mathcal{W} = [\mathcal{W}_1, \mathcal{W}_2, \ldots, \mathcal{W}_n]^T \in \mathbb{R}^{m \times n} \). This is usually what we can obtain using RBFNN in practice.

D. Fundamental Facts

Consider a nonlinear system:

\[
\dot{x}(t) = \ell(x(t))
\]

(11)

whose initial value is \( x_0 \). \( \ell(\cdot) \) represents a nonlinear mapping. System (11) can be discontinuous and assumed to have a unique solution in forward time for arbitrary initial states [18].

Lemma 1: [19] The origin of the nonlinear system (11) is said to be fixed-time stable with setting time \( T \) if there exists a Lyapunov function \( \mathcal{V}(x) \) and satisfies

\[
\dot{\mathcal{V}} \leq -\kappa \mathcal{V}^\gamma_1 - \zeta \mathcal{V}^\gamma_2 + \varpi
\]

(12)

where scalars \( \kappa, \zeta > 0, \gamma_1 \in (0, 1), \gamma_2 \in (1, \infty), \varpi \in (0, \infty) \). The convergence time \( T \) is bounded by

\[
T \leq \frac{1}{\kappa} \frac{1}{1 - \gamma_1} + \frac{1}{\zeta} \frac{1}{\gamma_2 - 1}
\]

(13)

Lemma 2: [20] For any \( x_i \in \mathbb{R}, i = 1, 2, \ldots, n \), the following inequalities hold:

\[
\sum_{i=1}^{n} |x_i|^{\gamma_1} \geq \left( \sum_{i=1}^{n} |x_i| \right)^{\gamma_1}, \quad \text{if } \gamma_1 \in (0, 1]
\]

(14)

\[
\sum_{i=1}^{n} |x_i|^{\gamma_2} \geq n^{1-\gamma_2} \left( \sum_{i=1}^{n} |x_i| \right)^{\gamma_2}, \quad \text{if } \gamma_2 \in (1, \infty)
\]

(15)

To facilitate the development of the following section, a new continuous function is defined as

\[
\text{sign}^\alpha(x) = |x|^\alpha \text{sign}(x)
\]

(16)

where \( \text{sign}(\cdot) \) is the standard signum and \( \alpha \geq 0 \) is a constant.
III. CONTROL SYSTEM DESIGN AND STABILITY ANALYSIS

A. Non-Singular Switching Manifold

Traditional TSM controllers face the singularity problem. Because the control law contains negative exponent of the state system, a singularity occurs when the state system reaches zero. In this paper, a non-singular switching manifold is introduced to avoid singularity.

Firstly, define two important parameters:

\[ \phi = \frac{-1}{\rho} \ln \vartheta \]

\[ \psi = \frac{\vartheta^{\rho-2}}{\rho - \vartheta \ln \vartheta} \]

Then, a nonlinear function \( h \) is introduced [14]:

\[ h = \begin{cases} 
\phi \text{sgn}^{\rho+1}(x) + \psi \vartheta^{\rho}|x|, & \text{if } |x| < \vartheta \\
\text{sgn}\theta(x), & \text{if } |x| > \vartheta
\end{cases} \]  

(17)

Take the first derivative of function (17), we have:

\[ \Xi = \begin{cases} 
(\phi + \vartheta) |x|^\rho + \psi(|x| \ln \vartheta + 1)\vartheta^{\rho}|x|, & \text{if } |x| < \vartheta \\
\rho |x|^\rho-1, & \text{if } |x| > \vartheta
\end{cases} \]  

(18)

where scalars \( \rho > 0, \vartheta \in (0, \exp(-1)) \) and satisfy the relationship \( \rho = 1 - \vartheta \). It is worth noting that the selection of \( \Phi \) and \( \Psi \) needs to guarantee the continuity of function \( h \) and \( \Xi \) when \( |x| = \vartheta \).

Define the tracking error as

\[ e := q_r - q \]  

(19)

where \( q_r \in \mathbb{R}^N \) is the desired position, \( e = [e_1, e_2, \ldots, e_N] \in \mathbb{R}^N \). Define the following vectors and matrices:

\[ H(e) = [h(e_1), h(e_2), \ldots, h(e_N)]^T \in \mathbb{R}^N \]

\[ \Gamma(e) = \text{diag}\{\Xi(e_i)\} \in \mathbb{R}^N \times N, \quad i = 1, 2, \ldots, N \]

\[ S \text{sgn}^\alpha(e) = [\text{sgn}^\alpha(e_1), \text{sgn}^\alpha(e_2), \ldots, \text{sgn}^\alpha(e_N)]^T \in \mathbb{R}^N \]

\[ D^{\rho-1}(e) = \text{diag}\{|e_1|^{\rho-1}, \ldots, |e_N|^{\rho-1}\} \in \mathbb{R}^N \times N, \quad i = 1, 2, \ldots, N \]

where \( \text{diag}\{\cdot\} \) denotes the diagonal matrix. Then, a non-singular switching manifold can be designed as [14]

\[ S = \hat{e} + K_1 H(e) + K_2 S \text{sgn}^\alpha(e) \]  

(20)

where \( S = [S_1, S_2, \ldots, S_N]^T \in \mathbb{R}^N \), constant \( \alpha > 1 \), matrices \( K_1, K_2 \in \mathbb{R}^{N \times N} \) are of positive diagonal definition. Upon differentiating \( S \) with respect to time, we can obtain:

\[ \dot{S} = \dot{\hat{e}} + K_1 \Gamma(e) \dot{\hat{e}} + \alpha K_2 D^{\rho-1}(e) \dot{\hat{e}} \]  

(21)

B. Non-Singular Fixed-Time Controller

Based on the designed switching manifold (20), a robust non-singular fixed-time controller is proposed in this section.

Substituting (21) and the second derivative of the tracking error (19) into the dynamic manipulator model (1), we can acquire

\[ \mathcal{M}_a \dot{S} = \mathcal{M}_a \dot{q}_r + \mathcal{M}_a K_1 \dot{\hat{e}} + \alpha \mathcal{M}_a K_2 D^{\rho-1}(\hat{e}) + C_a \dot{\hat{q}} + \mathcal{G}_a - \tau - \tau_e + d \]  

(22)

To simplify (22), we define a new variable \( \eta \), where

\[ \eta := S + \hat{q} \]  

(23)

Thus, the closed-loop error dynamic model (22) can be rewritten as

\[ \mathcal{M}_a \dot{S} + C_a S = \mathcal{M}_a \eta + \mathcal{C}_a \eta + \mathcal{G}_a - \tau - \tau_e + d \]  

(24)

Based on (24), we design the non-singular fixed-time controller as

\[ \tau_0 = K_3 S \text{sgn}^\beta_1(S) + K_4 S \text{sgn}^\beta_2(S) \]  

(25)

\[ \tau_1 = K_3 S \text{sgn}(S) + K_6 S \text{sgn}^0(S) \]  

(26)

\[ \tau = \mathcal{M}_a \eta + \mathcal{C}_a \hat{q} + \mathcal{G}_a - \tau_e + \tau_0 + \tau_1 \]  

(27)

where \( \tau_0 \) is the fixed-time sliding mode torque. The settling time of the sliding variable \( S \) to the origin is determined by the positive diagonal matrices \( K_3, K_4 \in \mathbb{R}^{N \times N} \) and the positive parameters \( \beta_1 \in (0, 1), \beta_2 \in (1, \infty) \) in \( \tau_0 \). \( K_5 \in \mathbb{R}^{N \times N} \) is the factor related to the uniformly ultimately boundedness (UBU) of the system. \( K_6 \in \mathbb{R}^{N} \) is the coefficient diagonal matrix of the robust term and its diagonal element satisfies \( K_{6i} \geq |E_1| \), where \( E = d + \epsilon M \hat{q} + C_1 \eta + C_2 \epsilon_1 \) and \( \epsilon M, C_1, C_2 \) are the approximation errors. \( \tau_e \) is directly acquired from the sensor. \( \mathcal{M}_a, \mathcal{C}_a \), and \( \mathcal{G}_a \) represent the estimation of \( \mathcal{M}_a, \mathcal{C}_a \) and \( \mathcal{G}_a \) using RBFNN, which can be expressed as \( \mathcal{M}_a = \hat{W}_M \mathcal{Y}_M, \mathcal{C}_a = \hat{W}_C \mathcal{Y}_C, \) and \( \mathcal{G}_a = \hat{W}_G \mathcal{Y}_G \). The updating law is designed as

\[ \dot{\hat{W}}_M = \mathcal{P}_M (\mathcal{Y}_M \dot{\hat{q}}_i - \epsilon_M \hat{W}_M) \]

\[ \dot{\hat{W}}_C = \mathcal{P}_C (\mathcal{Y}_C \dot{\hat{q}}_i - \epsilon_C \hat{W}_C) \]

\[ \dot{\hat{W}}_G = \mathcal{P}_G (\mathcal{Y}_G \dot{\hat{q}}_i - \epsilon_G \hat{W}_G), \quad i = 1, \ldots, N \]  

(28)

where \( \mathcal{P}_M, \mathcal{P}_C, \mathcal{P}_G \) matrices are positive diagonal definition, \( \epsilon_M, \epsilon_C, \epsilon_G \) are small positive constants for disturbance.

Under the control of (27), the system state can firstly reach the switching manifold (20) that is \( S = 0 \) within the fixed time \( T_1 \). Subsequently, along the switching manifold, the system state enters the small domain of the origin \( \Omega = \{e_i|e_i| \leq \vartheta\} \) within the fixed time \( T_2 \) and asymptotically gets to the origin in the end. According to Lemma 1, the fixed-time \( T_1 \) and \( T_2 \) can be calculated, which will be given in the next section.

C. Stability Analysis

Theorem 1: Upon introducing the non-singular fixed-time controller to the manipulator system (1), the tracking error of the system can achieve fixed-time convergence regardless of the initial conditions.

Proof: The fixed-time convergence analysis of the system can be divided into two stages. The first stage is to prove \( S = 0 \), where the system state reaches the switching manifold within \( T_1 \). The second stage is to prove that the tracking error \( e \) converges to the small domain \( \Omega \) within \( T_2 \), then asymptotically converges to the origin.

Stage 1:
Construct the Lyapunov function candidate as
\[
V = \frac{1}{2} S^T M_a S + \frac{1}{2} \sum_{i=0}^N W^T_{Mi} P_{Mi}^{-1} W_{Mi} + \frac{1}{2} \sum_{i=0}^N W^T_{Ci} P_{Ci}^{-1} W_{Ci}
\]
where \(W_{i}(\cdot)\) is the weight estimation error and \(\hat{W}_{i}(\cdot) = W^*_i - W_{i}(\cdot)\). Taking the derivative of (29), then substituting (2) and (24)–(28), we can obtain:
\[
\dot{V} = S^T \left[ -K_3 Sgn^{\beta_1}(S) - K_4 Sgn^{\beta_2}(S) - K_5 Sgn(S) \right] + S^T \mathcal{E} + \sum_{i=0}^N \varepsilon_{Mi} \dot{W}^T_{Mi} W_{Mi} + \sum_{i=0}^N \varepsilon_{Ci} \dot{W}^T_{Ci} W_{Ci} + \sum_{i=0}^N \varepsilon_{Gi} \dot{W}^T_{Gi} W_{Gi}
\]
Since
\[
\sum_{i=0}^N \varepsilon_{Mi} \dot{W}^T_{Mi} W_{Mi} \leq \sum_{i=0}^N \left( \varepsilon_{Mi}^2 / 2 \right) \| W^*_i \|^2 - \sum_{i=0}^N \varepsilon_{Mi}^2 / 2 \| W_{i}(\cdot) \|^2
\]
(30) can be transformed into
\[
\dot{V} \leq -\frac{N}{2} \lambda_{\text{max}} \left\{ K_3 \right\} \lambda_{\text{max}} \left\{ M_a \right\} \| S_i \|^2 - \frac{N}{2} \lambda_{\text{max}} \left\{ P_{Mi} \right\} \| W_{Mi} \|^2 - \frac{N}{2} \lambda_{\text{max}} \left\{ P_{Ci} \right\} \| W_{Ci} \|^2 - \frac{N}{2} \lambda_{\text{max}} \left\{ P_{Gi} \right\} \| W_{Gi} \|^2 + \xi_1 + \xi_2
\]
where \(\lambda_{\text{max}}\{\cdot\}\) and \(\lambda_{\text{min}}\{\cdot\}\) denote the largest and smallest eigenvalue of the matrix, respectively, \(\xi_1 = \sum_{i=0}^N \frac{\varepsilon_{Mi}^2}{2} \| W^*_i \|^2 + \sum_{i=0}^N \frac{\varepsilon_{Ci}^2}{2} \| W^*_i \|^2 + \sum_{i=0}^N \frac{\varepsilon_{Gi}^2}{2} \| W^*_i \|^2\), \(\varphi_1 = \min \left\{ \lambda_{\text{min}} \left\{ M_a \right\}, \lambda_{\text{max}} \left\{ P_{Mi}^{-1} \right\}, \lambda_{\text{max}} \left\{ P_{Ci}^{-1} \right\}, \lambda_{\text{max}} \left\{ P_{Gi}^{-1} \right\} \right\}\), \(\min\{\cdot\}\) represents the operation of taking the minimum value. According to Lemma 22 in [21], we can derive that
\[
V \leq V(0) e^{-\varphi_1 t} + \frac{\xi_1}{\varphi_1} \left( 1 - e^{-\varphi_1 t} \right), \quad t \geq 0
\]

Remark 1: For the Lyapunov function (29) \(V > 0\) and its first derivative satisfying \(\dot{V} \leq -\varphi_1 V + \xi_1\), we can acquire its maximal solution expressed as (32). Since \(V(0)\) is bounded, we can derive that \(S, \hat{W}(i), \text{ and } \hat{W}(i)\) are bounded.

Based on (30), we can develop
\[
\dot{V} \leq -S^T K_3 Sgn^{\beta_1}(S) - S^T K_4 Sgn^{\beta_2}(S) + \sum_{i=0}^N \left( -K_6 \right) \| \hat{W}_{Mi}(\cdot) \|^2 + K_7 \| \hat{W}(i) \|^2 + K_8 \| \hat{W}(i) \|^2 + K_9 \| \hat{W}(i) \|^2 + K_{10} \| \hat{W}(i) \|^2 + K_{11} \| \hat{W}(i) \|^2 + \xi_2
\]
(33) where \(\xi_2 = \sum_{i=0}^N \frac{\varepsilon_{Mi}^2}{2} \| W^*_i \|^2 + \sum_{i=0}^N \frac{\varepsilon_{Ci}^2}{2} \| W^*_i \|^2 + \sum_{i=0}^N \frac{\varepsilon_{Gi}^2}{2} \| W^*_i \|^2\).
Utilizing Lemma 2, \(V\) can be transformed into
\[
\dot{V} \leq -\frac{N}{2} \lambda_{\text{min}} \left\{ K_3 \right\} \lambda_{\text{min}} \left\{ M_a \right\} \| S_i \|^2 - \frac{N}{2} \lambda_{\text{min}} \left\{ K_4 \right\} \lambda_{\text{min}} \left\{ M_a \right\} \| S_i \|^2 - \frac{N}{2} \lambda_{\text{min}} \left\{ K_6 \right\} \| \hat{W}_{Mi}(\cdot) \|^2 - \frac{N}{2} \lambda_{\text{min}} \left\{ K_7 \right\} \| \hat{W}(i) \|^2 - \frac{N}{2} \lambda_{\text{min}} \left\{ K_8 \right\} \| \hat{W}(i) \|^2 - \frac{N}{2} \lambda_{\text{min}} \left\{ K_9 \right\} \| \hat{W}(i) \|^2 - \frac{N}{2} \lambda_{\text{min}} \left\{ K_{10} \right\} \| \hat{W}(i) \|^2 - \frac{N}{2} \lambda_{\text{min}} \left\{ K_{11} \right\} \| \hat{W}(i) \|^2 + \xi_2
\]
(34)
where \(\varphi_2 = \min \left\{ \lambda_{\text{min}} \left\{ K_3 \right\}, \lambda_{\text{max}} \left\{ M_a \right\}, \lambda_{\text{max}} \left\{ P_{Mi}^{-1} \right\}, \lambda_{\text{max}} \left\{ P_{Ci}^{-1} \right\}, \lambda_{\text{max}} \left\{ P_{Gi}^{-1} \right\} \right\}\). \(\xi_2\) can be proven to be bounded according to Remark 1.

The form of (34) is in keeping with Lemma 1. Therefore, the system state will arrive at the switching manifold (i.e. \(S = 0\)) within a designed settling time \(T_1\), where
\[
T_1 = \frac{1}{\varphi_2} \frac{2}{1 - \beta_1} + \frac{1}{\varphi_3} \frac{2}{\beta_1 - 1}
\]
(35)
stage 2:

In this stage, the system state moves along the switching manifold. According to (20), we can obtain \( S = \dot{e} + K_1 H(e) + K_2 \text{sgn}^\alpha(e) = 0 \).

When \( |e_i| > \vartheta \), the dynamics of the tracking error \( e_i \) becomes

\[
\dot{e}_i = -K_{1i} \text{sgn}^\rho(e_i) - K_{2i} \text{sgn}^\alpha(e_i)
\]

where \( K_{1i} \) and \( K_{2i} \) denote the diagonal elements of matrices \( K_1 \) and \( K_2 \), respectively. Define the Lyapunov function as \( V_1 = e_i^2 \). Taking the first derivative of \( V_1 \) yields:

\[
\dot{V}_1 = -2K_{1i}(e_i^2) - 2K_{2i}(e_i^2)
\]

It is obvious that (37) is consistent with (12), which means the tracking error \( e_i \) can converge to a small domain \( \Omega = \{ e_i | |e_i| \leq \vartheta \} \) within the designed settling time \( T_2 \) and

\[
T_2 = \frac{1}{K_{1i} - \rho + \frac{1}{K_{2i}} \alpha - 1}
\]

When \( |e_i| \leq \vartheta \), based on (17) the dynamics of the tracking error \( e_i \) changes to

\[
\dot{e}_i = -K_{1i} \left( \phi \text{sgn}^{\rho+1}(e_i) + \psi \text{sgn}^\alpha (e_i) \right) - K_{2i} \text{sgn}^\alpha(e_i)
\]

Construct the Lyapunov function candidate \( V_2 = \frac{1}{2} e_i^2 \). Differentiating \( V_2 \) with respect to time yields

\[
\dot{V}_2 = -K_{1i} |e_i| \left( \phi \text{sgn}^{\rho+2}(e_i) + \psi \text{sgn}^\alpha (e_i) \right) - K_{2i} |e_i| \text{sgn}^\alpha(e_i)
\]

Since \( 0 < \vartheta < \exp(-1) \), \( \rho = 1 - \vartheta \), we can derive that \( \phi > 0 \) and \( \psi > 0 \) based on (17). Therefore, we can obtain \( V_2 < 0 \), which means the system state can asymptotically converge to the origin. This completes the proof.

IV. SIMULATION

In this section, two simulation studies are given to prove the effectiveness of the proposed method. The first one is to compare the convergence effect of the proposed non-singular fixed-time controller and the general TSM controller in [9]. The other is to verify the performance of the manipulator when it interacts with the environment under the control of the proposed controller. The simulation plant is a two-joint manipulator shown as Fig.1, which is given by the parameters in the dynamics

\[
\mathcal{M}_a = \begin{bmatrix} \mathcal{M}_{11} & \mathcal{M}_{12} \\ \mathcal{M}_{21} & \mathcal{M}_{22} \end{bmatrix} \quad \mathcal{C}_a = \begin{bmatrix} \mathcal{C}_{11} & \mathcal{C}_{12} \\ \mathcal{C}_{21} & \mathcal{C}_{22} \end{bmatrix} \quad \mathcal{G}_a = \begin{bmatrix} \mathcal{G}_1 \\ \mathcal{G}_2 \end{bmatrix}
\]

where \( \mathcal{M}_{11} = L(1) + L(2) + 2L(3) \cos(q_2), \mathcal{M}_{12} = \mathcal{M}_{21} = L(2) + L(3) \cos(q_2), \mathcal{M}_{22} = L(2), \mathcal{C}_{11} = -L(3)q_2 \sin(q_2), \mathcal{C}_{12} = -L(3)(q_1 + q_2) \sin(q_2), \mathcal{C}_{21} = L(3)q_1 \sin(q_2), \mathcal{C}_{22} = 0, \mathcal{G}_1 = L(4)g \cos(q_1) + L(5)g \cos(q_1 + q_2), \mathcal{G}_2 = L(6)g \cos(q_1 + q_2), L = [2.9; 0.76; 0.87; 3.04; 0.87], g \) is the gravitational acceleration. To achieve fast convergence and small tracking error, the parameters in the control torque are finally tuned as \( \vartheta = 0.3, \rho = 0.7, \alpha = 1.9, K_1 = K_2 = K_5 = [5 0; 5 0], K_3 = [50 0; 50 0], K_4 = [2 0; 5 0], K_6 = [0.01 0; 0.01 0], \beta_1 = 0.5, \beta_2 = 2 \).

A. Comparison Study

This simulation setup is to track the same joint angle, where the desired joint angles are set as \( q_{r1} = 0.6 \) and \( q_{r2} = 1.7 \). The initial values of the joints are selected as \( q_1(0) = 1.1, q_2(0) = 1.5, \dot{q}_1(0) = \dot{q}_2(0) = 0 \). The curves of the angle tracking errors are shown in Fig.2, where the label \(-NS\) and \(-TSM\) in the legend represent the non-singular fixed-time controller and the general TSM controller, respectively. It is obvious that both controllers can converge the angle tracking error, and the system using the non-singular fixed-time controller has faster convergence effect than that using the general TSM controller.

B. Manipulator Interacting With Environment

In this simulation, the admittance model is utilized to modify the reference trajectory, and the proposed fixed-time controller is employed for angle tracking. The parameters in the dynamic model are approximated by BRFNN. An external torque will be exerted on the manipulator from sampling time 8 to 16, where \( \tau_e = [2; 0] \). The original reference trajectory is designed as \( q_{d1} = 1.25 - \frac{1}{2} \exp(-t) + \frac{7}{20} \exp(-4t), q_{d2} = 1.25 + \exp(-t) - \frac{1}{4} \exp(-4t) \). We set the external disturbance as \([2 \sin(0.2t); \cos(0.5t); 0.5 \sin(0.1\pi t)]\). The initial values of the joints are selected as \( q_1(0) = 0.6, q_2(0) = 1.8, \dot{q}_1(0) = \dot{q}_2(0) = 0 \). The parameters in the admittance model are chosen as \( \mathcal{D}_A = [3 0; 3 0], \mathcal{C}_A = [30 0; 30 0] \) and \( \mathcal{G}_A = [20 0; 20 0] \). By balancing the computing time and the approximate effect, the RBFNNs are designed with 5 neurons and their parameters are chosen as \( \sigma_\mathcal{M} = \sigma_\mathcal{C} = \sigma_\mathcal{G} = [-1; -0.5; 0; 0.5; 1] \), \( \mathcal{G}_4 = \mathcal{G}_5 = 10, \mathcal{E}_\mathcal{M} = 2, \mathcal{E}_\mathcal{C} = 0.55, \mathcal{E}_\mathcal{G} = 0.05, \mathcal{P}_{\mathcal{M}_{ii}} = 15, \mathcal{P}_{\mathcal{C}_{ii}} = 10, \mathcal{P}_{\mathcal{G}_{ii}} = 25 \). The simulation results are
shown as Figs. 3-5, where \( q_i \) is the real-time angle of the \( i \)-th joint of the manipulator, \( q_{di} \) represents the original reference joint angle, \( q_{ri} \) denotes the modified joint angle by admittance model. In the beginning, the modified trajectory gradually coincides with the reference trajectory, which is consistent with the admittance model that when \( f_{ext} = 0 \), \( q_r \rightarrow q_d \). Then, a large change of the modified trajectory occurs because of the external torque. The external torque is canceled after sampling time 16, therefore, the modified trajectory gradually coincides with the reference trajectory again. We set an input constrain as 50 \( \text{N-m} \) for safety, that is why the control torque is limited to 50 \( \text{N-m} \) at the beginning. During the whole process, the manipulator can quickly track the modified trajectory even though there is a large variation in the trajectory.

V. CONCLUSION

In this paper, we propose a non-singular fixed-time controller for unknown-dynamics manipulators. An admittance model is adopted into the system to realize compliant behavior of the manipulator while interacting with environment. The sliding mode control based on a non-singular fixed-time switching manifold is introduced to the system control design, which greatly improves the convergence effect of the system. In addition, RBFNNs are employed to approximate the unknown parameters in the dynamics. Simulation studies verify that the proposed controller has better tracking error convergence and faster response to external interaction than that using the traditional terminal sliding mode method.

REFERENCES

Zero-Shot Learning for Intelligent Fault Detection

Siyu Zhang and Hua-Liang Wei
Dept. Automatic Control and Systems Engineering
University of Sheffield
Sheffield, UK
E-mail: w.hualiang@sheffield.ac.uk (*Corresponding author)

Abstract—Signal-based fault detection, as an essential technology in many engineering and industrial applications, has received extensive attention. Nevertheless, in a real-world application setting, collecting samples for some specific types of faults may be time-consuming and damaging, implying that samples of unseen faults may be unavailable in the training step. Keeping this in mind, we present a new semantic space-based zero-shot learning (SSB-ZSL) method for fault detection. The implementation of the method comprises three phases: feature extraction, human-defined semantic space, and a feature embedding model to address this issue. To evaluate the proposed method, we carry out experiments on two real datasets: a vibration-based rolling element bearing dataset from Case Western Reserve University, and an acoustic-based air compressor fault dataset. The results are promising and significantly better than that given by the compared state-of-the-art methods, showing that the proposed method can effectively detect unseen types of faults in the absence of their samples.

Keywords—intricate fault detection; zero-shot learning; semantic description; deep learning; 1D CNN

I. INTRODUCTION

One-dimensional (1D) signals and time series are a common data format for presenting and monitoring system or process operation status. The procedure of traditional data-driven fault detection approaches includes three steps: data acquisition and pre-processing, feature extraction or selection, and fault classification [1].

Feature extraction, as a key step, strongly affects fault detection performance. Roughly, feature extraction methods can be divided into two groups: traditional matrix decomposition (transformation) and convolutional neural networks (CNNs). Commonly used linear transformation and dimensionality reduction methods include principal component analysis (PCA) [2] and independent component analysis (ICA) [3]. The commonly used non-linear transformation approach is the kernel method including kernel principal analysis (KPCA) [4]. Recently, with the advancement of machine learning especially complex neural network techniques, CNNs have wide applications in the field of fault detection and diagnosis. It is known that for a wide range of applications in e.g. industrial machines, data are recorded as 1D signals or time series, typical CNNs which were initially designed for 2D data processing, may not be applicable. To make use the good properties of CNNs, Peng et al.[5] proposed a deeper 1D CNN for fault detection of wheelset bearings.

After feature extraction, the resulting features are put into a fault classification model to determine the machine’s operational status, e.g., heathy or faulty, and the fault types. However, the classical three-step defect detection procedure has several drawbacks and limits. First, in implementing the three-step procedure, it is assumed that fault samples are sufficiently available for model training; however, in practice there are many cases where no or just few data may be available for unseen faults during everyday process operation in the industrial settings [6]. Second, because some specific types of faults can be harmful and cause significant losses, few plants would be allowed to operate to failure or large-scale problems and collect instances to train a fault detection model [7]. Third, machines often deteriorate from good health to failure over time, as a result gathering appropriate fault samples for data-driven techniques is time-consuming and expensive.

Zero-shot learning (ZSL), which was firstly proposed by Lampert et al. [8], has lately received significant attention. It aims to detect the unseen objects using a high-level human-defined description rather than utilizing trained objects. The high-level descriptions are semantic attributes, such as color, shape, or geographic information, which could be perceived and extracted from seen and unseen label information rather than unseen samples. Hence, ZSL is a model for classifying unseen classes without using all of their samples during the model training stage[9]. But in the meantime, the side information shared between seen (training) and unseen (testing) classes is used during the training and testing stages [10]. In this article, we focus on inductive models when designing the ZSL framework for fault detection. Classification approaches under the inductive setting could be divided into three parts: direct-attribute prediction, semantic space embedding, and sample generation [11].

The semantic space model maps learned visual features to semantic space [12, 13]. Frome et al. [14] presented a deep visual-semantic embedding model to classify the unseen samples with labeled image and side information gleaned from both seen and unseen labels. Akata et al. [15] offered a bi-linear function that determines if an image feature and a label embedding are compatible. The parameters of this function are learned from training data (the seen instances) to ensure that correct classes get higher scores than the wrong ones when given an image. Romero et al. [16] proposed an approach based on a general framework, which models the connections between visual features, semantic attributes, and classes as a two linear layers neural network. Kodirov et al. [17] took a semantic
encoder-decoder to embed an image feature vector into the semantic space and then reconstruct the original feature.

As an inspiration, ZSL could bring a breakthrough in intelligent fault detection, especially for detecting the unseen faults without using their samples in training the model. There exist some preliminary research results in using ZSL for fault detection. For example, Feng et al. [6] proposed a fault description model to detect unseen faults in complex mechanical systems based on an attribute transfer strategy. Xing et al. [18] presented a label description space embedding model for detecting the unseen compound faults of machines. Xu et al. [19] used a visual space based model to detect unseen compound faults of devices.

Building appropriate semantic attributes for vibration-based faults is essential since semantic attributes in image recognition are unsuitable for vibration signals. It should be noticed that when a new class of fault occurs in a system, what we will see is the fault description rather than fault samples. For example, from the description “A pipeline accessory for opening and closing the pipeline, controlling the flow direction, regulating and controlling the parameters of the transported medium,” workers can detect the object “valve” without seeing it at all. We all know that “leakage inlet valve” that happened in the valve based on the description of the valve. Furthermore, the attributes could be shared among multiple classes of seen or unseen faults. For example, the high-level description “leakage fault” is shared with the fault of “leakage inlet valve” and “leakage outlet valve”. Then the fault attributes could be directly transferred to unseen types of faults in the testing stage. Finally, the fault attributes could comprise a variety of factors, such as the fault's location, the connected process variable, the fault's size, etc. The fault attributes give side information for unseen faults, allowing the model to detect unseen faults and solve the zero-shot fault detection problem directly.

Inspired by the ZSL scheme reported in [15], this article proposes a new semantic space based zero-shot learning (SSB-ZSL) for fault detection method for the bearings and air compressor. The implementation procedure of SSB-ZSL is as follows. The first step is feature extraction with 1D CNN in the training stage. The second step is to establish a semantic space of fault attributes, shared between seen faults and unseen faults. In the third step, it learns a compatibility between fault features and fault attributes by a bi-linear compatibility function to find the highest-ranking unseen fault type. Finally, in the testing stage, it feeds the unseen class faults into the trained model with the best parameter and then match the highest-ranking fault type. Two case studies are designed and experiments are carried out based on two datasets for: 1) rolling bearings, and 2) air compressors. Both covering comprehensive range of fault types.

The remainder of the article is organized as follows. Section II presents the details of the SSB-ZSL model. Section III utilizes two cases to verify the effectiveness of the proposed method. Finally, Section IV concludes this paper.

II. THE PROPOSED METHOD

This section introduces the proposed method SSB-ZSL with problem formulation and model structure.

A. Problem formulation

Assume that there is a set of seen fault data $S = \{(x^s_i, y^s_i)\}_{i=1}^{N_s}$, with $x^s_i \in X^s$, $y^s_i \in Y^s$, which contains $N_s$ fault samples and $s$ types of seen faults. Each sample $x^s_i$ corresponds to a label $y^s_i$. Likewise, given a testing (unseen) dataset $U = \{(x^u_i, y^u_i)\}_{i=1}^{N_u}$ with $x^u_i \in X^u$, $y^u_i \in Y^u$, the dataset consists of $N_u$ fault data samples and $u$ classes of unseen faults. Each sample $x^u_i$ corresponds to a label $y^u_i$. The fault attributes of a fault are denoted as $A = [A^s, A^u] \in R^{L \times C}$, where $L = s + u$, and $C$ is the number of fault attributes. It is essential to point out that both $A^s$ and $A^u$ are available in the training stage because the fault attributes are prior knowledge that is class-level common knowledge. Therefore, we can obtain the fault attributes in advance. The samples and classes need to meet the following conditions in ZSL settings: $Y^s \cap Y^u = \emptyset$.

B. Model structure

The proposed SSB-ZSL consists of three stages: feature extraction, semantic space definition, and feature embedding. The model structure is depicted in Fig.1.

- **Start**
  - Build the model
  - Initialize parameters
  - Train model and update $W$

- **Maximum number of iterations**
  - Yes: Finish training
  - No: Continue training

- **Trained model**
  - Classify the unseen faults

**SSE-ZFD model**

- Raw data
- Feature extraction
- Training features
- Validation features
- Measurement of compatibility
- Human-defined semantic space
- Fault 1
- Fault 2
- Fault u

**Figure 1.** The SSB-ZSL model structure.

(i). Feature extraction

We propose a 1D CNN network and use it as a feature extractor for fault detection. The architecture of the designed 1D CNN is shown in Table I. It contains two convolution layers, two max-pooling layers, one flatten layer, and one fully-connected layer. The input of the 1D CNN is the 1D time-series (signal vectors), and the output from the fully-connected layer is extracted feature $\theta(x)$ for each sample.
TABLE I.  ARCHITECTURE OF 1D CNN

<table>
<thead>
<tr>
<th>Layer Name</th>
<th>Description</th>
<th>Kernel Size</th>
<th>Stride</th>
<th>Kernel Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>Convolution</td>
<td>1×3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>P1</td>
<td>Max-pooling</td>
<td>1×4</td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>C2</td>
<td>Convolution</td>
<td>1×3</td>
<td>3×1</td>
<td></td>
</tr>
<tr>
<td>P2</td>
<td>Max-pooling</td>
<td>1×4</td>
<td></td>
<td>128</td>
</tr>
<tr>
<td>F</td>
<td>Flatten</td>
<td>1×128</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>FC</td>
<td>Fully-connected</td>
<td>1×64</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

(ii). Second-stage neural network sub-model

Semantic space builds a bridge between seen faults and unseen faults in zero-shot learning for fault detection. Here, we use fault attributes to construct the semantic space. The fault attributes usually contain fault position and fault effect, which can easily be perceived without seeing the unseen samples. Fault attributes are usually specified as computer-readable vector forms [20], and each attribute could be a binary value \( \mathbf{q}_{x,1} \in \{0,1\} \) or a continuous value \( \mathbf{q}_{c} \in [0,1] \). The attributes for each type of fault can be defined as:

\[
\Phi(y) = [\mathbf{q}_{y,1}, \ldots, \mathbf{q}_{y,E}]^T
\]

where \( \mathbf{q}_{x,1} \) could be one of the two binary numbers \( \{0,1\} \) or a real number between 0 and, \( y \) denotes fault class, and \( E \) denotes the dimension of attributes for a fault class.

(iii). Feature embedding

Following [15], we derive a predictive function by maximizing the compatibility \( F \) as follows:

\[
f(x; w) = \max_{y \in \mathcal{Y}} F(x, y; w)
\]

where \( w \) denotes the parameter vector of \( F \) and can be written as a \( D \times E \) matrix \( W \) with \( D \) being the extracted features dimension and \( E \) being the attributes dimension. The bi-linear compatibility function \( F: \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R} \) between a raw fault data space \( \mathcal{X} \) and a fault label space \( \mathcal{Y} \) can be defined as:

\[
F(x, y; W) = \theta(x)^T W \Phi(y)
\]

where \( \theta(x) \) stands for feature extraction procedure while \( \Phi(y) \) stands for fault semantic space. \( F(x, y; W) \) is a ranking-based compatibility function, which provides the proper label with the highest rank over all other labels. The above concept is similar to the web scale annotation by image embedding (WSABIE) algorithm [21], but there is an obviously significant difference between WSABIE and our method in that the former learns both \( \Phi(y) \) and \( W \), whereas the later simply learns \( W \) and uses \( \Phi(y) \) as fault attributes.

(iv). Parameter learning

Motivated by the unregularized structure SVM formulation [22], the objective function is defined as:

\[
\frac{1}{N} \sum_{n=1}^{N} \max_{y \in \mathcal{Y}} \{0, r(x_n, y_n, y)\}
\]

where the loss function \( r(x_n, y_n, y) \) takes the form:

\[
\Delta(y_n, y) + F(x_n, y; W) - F(x_n, y_n; W)
\]

Here, we choose \( \Delta(y_n, y) = 1 \) as margin. We use only samples from seen faults to learn the best \( W \) in the training stage, so we use stochastic gradient descent (SGD) to optimize \( W \) to find the highest ranked fault type \( y \). If \( \arg \max_y r(x_n, y_n, y) \neq y_n \), we optimize \( W \) as follows:

\[
W^{(i+1)} = W^{(i)} - \eta_i (\Phi(y_n) - \Phi(y))
\]

where \( \eta_i \) is the learning rate used at the \( i \)-th iteration. In the test stage, we map the extracted feature onto the best \( W \) and search for the nearest fault attribute vector, which belongs to one of the unseen fault classes, using the cosine similarity.

III. EXPERIMENTS

In this section, two datasets are discussed, including rolling bearing fault dataset Case Western Reserve University (CWRU) [23] and reciprocating air compressors fault data [24].

A. Case study I - CWRU dataset

(i). Introduction to the dataset

The Case Western Reserve University Bearing Data Center has a CWRU dataset with vibration-based rolling bearing fault data. A 2 hp reliance electric motor, a torque, and a dynamometer are all part of the test bench. In addition, at the fan-end and drive-end, an acceleration sensor is mounted above the bearing housing to record vibration acceleration when collecting fault data. Inner race fault, rolling element fault, and outer race fault with four working loads (0,1,2 and 3 hp) being positioned on the drive-end bearing and fan-end bearing, respectively. Using electro-discharge machining (EDM), each class of defect has a fault diameter ranging from 0.007” to 0.028” (inches) on the bearings.

TABLE II. FAULTS LABELS (9 FAULTS IN EACH WORKING LOAD)

<table>
<thead>
<tr>
<th>Fault Number</th>
<th>Fault Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.007” fault in inner race (0 – 3 hp)</td>
</tr>
<tr>
<td>2</td>
<td>0.007” fault in rolling element (0 – 3 hp)</td>
</tr>
<tr>
<td>3</td>
<td>0.007” fault in outer race (0 – 3 hp)</td>
</tr>
<tr>
<td>4</td>
<td>0.014” fault in inner race (0 – 3 hp)</td>
</tr>
<tr>
<td>5</td>
<td>0.014” fault in rolling element (0 – 3 hp)</td>
</tr>
<tr>
<td>6</td>
<td>0.014” fault in outer race (0 – 3 hp)</td>
</tr>
<tr>
<td>7</td>
<td>0.021” fault in inner race (0 – 3 hp)</td>
</tr>
<tr>
<td>8</td>
<td>0.021” fault in rolling element (0 – 3 hp)</td>
</tr>
<tr>
<td>9</td>
<td>0.021” fault in outer race (0 – 3 hp)</td>
</tr>
</tbody>
</table>

In this study, recordings for the case of 12kHz drive-end bearing fault are chosen as our experimental dataset. Overall, there are 4 group of experiments, including 0 – 3 hp. For each group, there are nine kinds of faults in total. Each type of fault takes the first 102,400 data for overlap sampling, and the overlap ratio is 50%, so the length of each sample is 1024, and each class has 200 samples. The details of faults in the dataset are summarized in Table II. The training data belong to 0.007” and 0.014”, and we focus on detecting a type of larger size faults, i.e., 0.021” faults.

(ii). Model implementation

The first step of SSD-ZSL is feature extraction. A 1D CNN model was developed and utilized to extract fault features from raw vibration signals; the input to the 1D
CNN is of $1 \times 1024$ and the output is of $1 \times 64$ from the fully connected layer. The second step is to build a human-defined semantic space, i.e., fault attribute matrix, as shown in Fig. 2. The details of each attribute are shown in Table III. Note that the attribute matrix does not distinguish the same type of faults with different working loads. For example, 1 hp 0.007" fault in inner race and 2 hp 0.007" fault in inner race has the same attributes. Also note that in the training stage, we just need the fault attributes of 0.021" fault detection but do not need its samples. In fact, once the fault attributes are obtained, we could remove the unseen fault samples from the training data.

![Fault attributes matrix A for CWRU.](image)

**TABLE III. FAULT ATTRIBUTES FOR CWRU**

<table>
<thead>
<tr>
<th>Attribute Number</th>
<th>Fault attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Att1</td>
<td>Centered position</td>
</tr>
<tr>
<td>Att2</td>
<td>Occurred at inner ring</td>
</tr>
<tr>
<td>Att3</td>
<td>Occurred at outer ring</td>
</tr>
<tr>
<td>Att4</td>
<td>Occurred at rolling element</td>
</tr>
<tr>
<td>Att5</td>
<td>0.007&quot; failure diameter</td>
</tr>
<tr>
<td>Att6</td>
<td>0.014&quot; failure diameter</td>
</tr>
<tr>
<td>Att7</td>
<td>0.021&quot; failure diameter</td>
</tr>
</tbody>
</table>

There are 36 types of bearing faults in total, which are divided into four groups, and each group has nine types of faults. As for dataset split, four types of 0.007" and 0.014" faults are in the train sets, two types of 0.007" and 0.014" faults are validation sets, and the rest three types of 0.021" faults belong to the test set. The number of training instances is $200 \times 4 = 800$, the number of validation instances is $200 \times 2 = 400$, and the number of test instances is $200 \times 3 = 600$.

As for evaluation criteria, we are interested in the accuracy of each type of unseen faults, so we measure the average per-class top-1 accuracy as follows [13]:

$$\text{acc} = \frac{1}{|Y^u|} \sum_{y^u} \frac{\text{correct detections in } y^u}{\text{samples in } y^u}$$  \hspace{1cm} (7)

where $y^u$ is an unseen large size fault detected and $Y^u$ is the number of unseen large size faults.

(iii). Fault detection accuracy

The data split for the train/validation/test set and the results of SSB-ZSL is presented in Table IV. With diverse working loads ranging from 0 hp to 3 hp, the accuracies range from 66.67% to 87.67%. Our results are far higher than the random level of 33.33%, suggesting that fault attributes shared by seen small-size faults and unseen large-size fault can be used to detect unseen large-size faults without their samples in the training stage. When it comes to the results of each group, the accuracies for the 0 hp and 2 hp working load groups outperform that for the 1 hp and 3 hp working load groups. This is because the inner race and outer race fault signals appear to be identical in the frequency domain, making them more difficult to distinguish than the rolling element faults. In the following comparative studies, we will show that our results are pretty competitive in comparison other state-of-the-art methods.

**TABLE IV. DATA SPLIT FOR CWRU**

<table>
<thead>
<tr>
<th>Group</th>
<th>Training Faults</th>
<th>Validation Faults</th>
<th>Test Faults</th>
<th>Accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 hp</td>
<td>1, 2, 4, 6</td>
<td>3, 5</td>
<td>7, 8, 9</td>
<td>78.50</td>
</tr>
<tr>
<td>1 hp</td>
<td>1, 2, 4, 6</td>
<td>3, 5</td>
<td>7, 8, 9</td>
<td>66.67</td>
</tr>
<tr>
<td>2 hp</td>
<td>1, 2, 4, 6</td>
<td>3, 5</td>
<td>7, 8, 9</td>
<td>87.67</td>
</tr>
<tr>
<td>3 hp</td>
<td>1, 2, 4, 6</td>
<td>3, 5</td>
<td>7, 8, 9</td>
<td>66.78</td>
</tr>
</tbody>
</table>

(iv). Comparison with other ZSL methods

We used the same setting to examine different ZSL approaches mentioned in Section I, namely SJE, DEVISE, and SAE. It is worth noting that all the three approaches were designed for image recognition, and the visual features they offer are not helpful for fault detection here. So, we applied our proposed fault attributes in these methods. Meanwhile, ZSL nearly always uses deep CNNs, such as GoogleNet [25] and SIFT [26], to extract features. SJE, DEVISE and SAE were trained to learn from raw signals. The results are displayed in Table V.

From the results, it is clear that our method shows far better performance than these three compared semantic space embedded ZSL algorithms. This may be explained that these three methods were initially developed for image recognition, however, image features and visual attributes are unavailable for industrial vibration signals, therefore their accuracies suffer significantly. From another aspect, these results demonstrate how important it is to choose a suitable feature extraction method for fault detection based on 1D signals. Traditionally, most applications of ZSL are limited to pattern limited to pattern recognition, where 2D visual features are usually needed; few work has been done to deal with 1D time series (e.g., vibration signals). This motivates us to contrive a new reasonable strategy for zero-shot fault detection by combining 1D CNN and ZSL.

**TABLE V. COMPARISON WITH ZSL METHODS FOR CWRU (%)**

<table>
<thead>
<tr>
<th>Group</th>
<th>0 hp</th>
<th>1 hp</th>
<th>2 hp</th>
<th>3 hp</th>
</tr>
</thead>
<tbody>
<tr>
<td>SJE</td>
<td>38.17</td>
<td>39.17</td>
<td>32.17</td>
<td>39.50</td>
</tr>
<tr>
<td>DEVISE</td>
<td>39.67</td>
<td>38.83</td>
<td>29.33</td>
<td>40.50</td>
</tr>
<tr>
<td>SAE</td>
<td>24.17</td>
<td>31.50</td>
<td>26.67</td>
<td>33.17</td>
</tr>
<tr>
<td>SSB-ZSL</td>
<td>78.50</td>
<td>66.67</td>
<td>87.67</td>
<td>66.78</td>
</tr>
</tbody>
</table>

B. Case study II - Air compressor fault dataset

(i). Introduction to the dataset

Verma et al. [27] collected the reciprocating air compressor fault data [24], containing 1 healthy state and 7 different faults. All acoustic signals were acquired for a period of 5 seconds at a 50kHz sampling rate. All the data were pre-processed to lessen the impact of noise and...
outliers by filtering, clipping, smoothing, and normalization.

In this case study, the following 6 types of faults are considered in our experiments: bearing fault, flywheel fault, leakage inlet valve fault, leakage outlet valve, non-return valve fault, and piston ring fault. Overall, there are 3 groups of experiments to detect unseen types of faults. Following the data pre-processing method used in Case Study I, Section III, the dataset used in this study also has 200 samples for each type of fault, and each sample has 1024 lengths of data. The details of faults in the dataset are displayed in Table VI.

<table>
<thead>
<tr>
<th>Fault Number</th>
<th>Fault Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bearing fault</td>
</tr>
<tr>
<td>2</td>
<td>Flywheel fault</td>
</tr>
<tr>
<td>3</td>
<td>Leakage inlet valve fault</td>
</tr>
<tr>
<td>4</td>
<td>Leakage outlet valve fault</td>
</tr>
<tr>
<td>5</td>
<td>Non-return valve fault</td>
</tr>
<tr>
<td>6</td>
<td>Piston ring fault</td>
</tr>
</tbody>
</table>

(ii). Model implementation

First, we used the same implementation procedure as for the CWRU dataset. The second step is building human-defined semantic space, i.e., fault attribute matrix, as shown in Fig.3, and the details of each attribute are shown in Table VII. Then for each group, the data are split into three parts: 3 types of fault data for training, 1 type of fault data for validation, and 2 types of fault data for test. The number of training instances is 200×3=600, the number of validation instances is 200×1=200, and the number of test instances is 200×2=400. The evaluation criteria are the same as for CWRU dataset.

![Figure 3. Fault attributes matrix A for air compressor.](image)

<table>
<thead>
<tr>
<th>Attribute Number</th>
<th>Fault Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Att1</td>
<td>Result in wear on</td>
</tr>
<tr>
<td>Att2</td>
<td>Result in to crack</td>
</tr>
<tr>
<td>Att3</td>
<td>Leakage fault</td>
</tr>
<tr>
<td>Att4</td>
<td>Result in air leakage</td>
</tr>
<tr>
<td>Att5</td>
<td>Damage fault</td>
</tr>
<tr>
<td>Att6</td>
<td>Related to rotating equipment</td>
</tr>
<tr>
<td>Att7</td>
<td>Occurred at inlet</td>
</tr>
<tr>
<td>Att8</td>
<td>Occurred at outlet</td>
</tr>
</tbody>
</table>

(iii). Fault detection accuracy for the air compressor

As can be seen from the results presented in Table VIII, the accuracies vary from 73.5% to 78.00%, which are consistent among the 3 groups. The accuracy details of each unseen fault are presented in the confusion matrices in Fig.4, where it can be noticed that for all types of faults (but fault 5), the accuracies far exceed the random level of 50.00%, and for fault 5 it is 56.5%. Fault 5 is more likely to be classified as fault 6. From the attribute matrix in Fig.3, fault 5 and fault 6 both have att4, and fault 5 has att5 additionally. It is clear that fault 3 and fault 4 also share att5, which are used for training in Group C. In the training stage, att5 is well studied and has relatively low discriminability. In the test stage, att4, which has never been learned, is more discriminatory to distinguish fault 6 and misclassify fault 5. Unfortunately, there is only a relatively small description of piston ring fault (fault 6) in [24], which means that it is difficult to define comprehensive attributes for fault 6. It will be helpful if more information can be available for fault 6, and more attributes could be shared with other faults. So, it is challenging to perfectly distinguish fault 5 and 6 based on the similarity of attributes. Besides, due to the nature of the zero-shot problem, it is more difficult to train a model using the seen faults with only a few fault attributes to detect the unseen faults with much more fault attributes. Furthermore, since we used human-defined attribute matrix rather than unsupervised text corpora, we have to acknowledge the disadvantage that it is difficult to classify accurately when faced with insufficient priori knowledge. But for fault detection problem, making full use of interpretability and human priori knowledge in this field are more beneficial for accuracy.

![Figure 4. Confusion matrices of the results of unseen faults.](image)

(iv). Comparison with other feature extraction methods

We focus on three commonly used traditional feature extraction methods, namely, PCA, ICA and KPCA, and compare their performance with the proposed 1D CNN. We implement the experiments by using the scikit-learn package [27] for these three methods. As for parameters, each method adopts 64 components extracted from the raw data.
signal vectors of length 1024. The accuracies of different feature extraction methods are presented in Table IX. In all of the groups, we achieve very impressive results. For Group C, the proposed 1D CNN achieves a slightly better performance than KPCA. However, 1D CNN obtains significant improvements for Group A (30.75%) and Group B (24.00%) compared with the best results of the other three methods. In this case, it can be concluded that our approach outperforms the traditional feature extraction approaches. This may be explained as follows. Firstly, PCA, ICA and KPCA, due to their nature and implementation mechanism, may well tackle linear and some certain types of non-linear problems. CNN, as an artificial neural network with several or many hidden layers, however, could solve both complicated linear and very complex non-linear problems. Secondly, multi-layer CNNs, with non-linear functions as activation functions, could have better performance. Although overfitting may exist, it can be resolved via a regularisation scheme. These analyses suggest that the designed 1D CNN, together with ZSL, can well handle 1D signals and achieve remarkable and significant improvement in fault detection.

<p>| TABLE IX. COMPARISON WITH FEATURE EXTRACTION METHODS |</p>
<table>
<thead>
<tr>
<th>Group</th>
<th>PCA (%)</th>
<th>ICA (%)</th>
<th>KPCA (%)</th>
<th>SSB-ZSL (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>44.75</td>
<td>46.50</td>
<td>44.25</td>
<td>77.25</td>
</tr>
<tr>
<td>B</td>
<td>44.25</td>
<td>54.00</td>
<td>48.75</td>
<td>78.00</td>
</tr>
<tr>
<td>C</td>
<td>61.50</td>
<td>62.00</td>
<td>69.00</td>
<td>73.50</td>
</tr>
</tbody>
</table>

IV. CONCLUSION

This main contribution of the article is the proposal of a novel SSB-ZSL method for zero-shot fault detection, which only uses the seen types of faults in the training stage to classify the unseen types of faults in the testing stage. We consider two common applications in industry: bearing fault and air compressor fault detection. For the first case, we focus on detecting unseen large size faults from the seen small size of faults. The results show that the proposed method significantly performs better than other three compared ZSL methods. For the second case, we classify the unseen types of faults from seen types of faults and obtain much better results than the other three traditional feature extraction methods. Given the fact that in practical industrial scenarios, it is difficult to collect specific types of fault signals and existing methods do not work well for fault detection for such cases, the results from this study may shed light on finding a solution.

ACKNOWLEDGMENT

The authors acknowledge that this work was supported in part by NERC (Ref. NE/V001787/1), NERC (Ref. NE/V002511/1) EPSRC (Ref. EP/H00453X/1), EPSRC (Ref. EP/I011056/1), EU Horizon 2020 (Ref. 637302). Siyu Zhang gratefully acknowledges a scholarship from the Chinese Scholarship Council (CSC).

REFERENCES


Implicit Representation of Single-view Reconstruction For Texture-less 6DoF Pose Estimation

Zian Zheng, Yong Wu, Tao Wang, Chong Chen
Guangdong Provincial Key Laboratory of Cyber-Physical System
Guangdong University of Technology
Guangdong Province, China
Wangtao_cps@gdut.edu.cn

Abstract—The six Degree-of-Freedom (6DoF) pose estimation is a critical technology in the flexible operation of industrial robots and plays a significant part in the construction of vision systems such as robot grasping and assembly. The task of estimating the texture-less objects’ poses in complicated industrial environments is difficult because texture-less objects lack enough texture information to extract enough feature points for recognition. To address the challenge of 6DoF pose estimation under severe occlusion and texture-less, a novel convolutional neural network, IRNet, is proposed in this study, which is founded on single-view reconstruction for implicitly representing 6DoF pose. In the experiment, a set of high-precision texture-less bottle datasets were constructed for pose estimation in industrial scenes. The experiments demonstrate our method’s superiority and robustness compared with existing techniques.

Keywords—Texture-less; 6DoF pose estimation; Object recognition

I. INTRODUCTION

An object’s 6DoF pose consists of its 3D position and 3D orientation, making it appropriate for robot grasping and augmented reality. However, there are a lot of challenges with 6DoF pose estimation, such as background clutter and insufficient information. The requirement to learn, detect, and reliably localize texture-less rigid objects from photos emerges in a number of applications since they are common in human settings, particularly in industrial scenarios [1]. However, the lack of texture in texture-less objects prevents better feature extraction and recognition, resulting in less significant or less generalized 6DoF pose estimation for texture-less objects in some industrial scenes.

Many methods have been proposed in order to improve the performance of 6DoF pose estimation and to extend its application areas. A traditional class of algorithms are the non-learning-based approaches, both feature-based and template-based techniques are included. As for feature-based methods [2], they take the feature points from the source image, construct local descriptors, and match this set of features with the database to find correspondences. However, the texture-less object usually does not provide an adequate number of descriptive feature points. Template-based approaches [3] use rigid templates with specified 6DoF poses, compare them with object regions of different scales and orientations, trying to maximize the similarity score to find the best match. However, these methods can not deal with the problem of occlusion. Moreover, both the non-learning-based approaches have low accuracy and poor real-time in cluttered scenes [4].

For texture-less objects, the traditional method extracts feature points using common feature descriptors and then matches the feature points with the model of the texture-less object in the scene to obtain the 6DoF pose. The method is often not highly discriminative and cannot establish a stable feature point matching relationship from the scene to the model.

The approaches based on deep learning have gained popularity in the last decade as a result of the development of machine learning and the accessibility of high-quality datasets. In comparison to non-learning-based approaches, many academics have used deep learning-based algorithms for 6DoF pose estimation and have obtained better and more reliable results. Inspired by the YOLO network, Tekin et al. [5] directly predicted the 2D image coordinates of the projected points of the object’s 3D bounding box and then estimated the 6DoF pose of the object using a Perspective-n-Point (PnP) algorithm [6]. However, this method didn’t address texture-less and occluded objects. Peng et al. [7] presented PVNet, which used CNN to predict label information for the object and unit vectors indicating the direction from every pixel to all key points, as well as confidence scores. However, for texture-less objects, the approach is poorly robust and introduces the symmetry ambiguity problem.

The key contributions of this paper are the following two components, which help with the aforementioned issues:

a) In order to apply texture-less object pose estimation to realistic industrial scenarios, a texture-less bottle dataset for industrial sorting is presented, which contains four bottles without discriminative color and texture. Meanwhile, a synthetic dataset for training our 6DoF pose estimation network was constructed.

b) A new convolutional neural network, IRNet, based on single-view reconstruction for implicitly representing 6DoF pose is proposed, which only needs to generate images as supervised signals by sampling CAD model views without a specific reconstruction model.
II. RELATED WORK

The correspondence-based methods: When a depth image is available, it is possible to convert the observed depth image into a point cloud (PC) by combining the internal parameters of the depth camera, which need to be aligned with the existing full 3D PC. Global alignment is required when there are large deviations between the single-view PC and the complete 3D PC, while local alignment can be used when the single-view PC and the complete 3D PC are close to each other. The global alignment techniques are primarily separated into two types: one is based on RANSAC-type methods such as Super 4PCS [8], where three or four pairs of points are randomly selected for voting and the best correspondence is selected as the final transformation; the other is based on feature descriptor methods, where significant feature points are extracted separately from the PC. Common 3D descriptors like FPFH [9], Spin image [10], and SHOT [11] can be used to find the correspondence between different parts of the 3D PC and the target 3D PC for global alignment. Then, it is iteratively optimized using the Iterative Closest Point (ICP) technique [12]. In 2017, Wong et al. [13] introduced a technique for obtaining target object poses that combines RGB-based object segmentation and partial alignment of depth images. They proposed a new metric for rating the quality of model alignment and performed multi-hypothesis alignment to achieve accurate pose estimation with a position error of one cm and an error of fewer than five degrees.

The template-matching methods: The primary goal of the template matching-based method is to solve the problem of the poor performance of correspondence-based methods on texture-less objects. It is mainly used to identify the poses of texture-less objects. For weakly textured objects, traditional methods rely mainly on the technique of template matching. The method of Hinterstoisser et al. [14] is the classical template matching method, which gives a relatively robust estimation result even when the object is under-textured. However, using as many templates as possible to cover all potential viewing points is necessary to estimate the target object's 6DoF pose more precisely, which increases computational work and makes meeting the requirements of real-time detection unfeasible.

In deep learning, it is generally difficult to extract feature points with template-based methods, and all that can be used is the information from RGB images. Therefore, the 3D model is usually projected from different angles to obtain a rendered image of the model at each angle, and the corresponding poses are recorded and used as templates. Hodaš et al. [15] introduced a technique for multimodal template matching using RGB-D images by matching feature points in different modes to select with the corresponding template, and then using the pose associated with the template as the initial pose, which is then further optimized using the ICP algorithm. Muñoz et al. [16] presented a method to estimate a given texture-less object's pose. The approach first determines the Euclidean distance between the corresponding edge of the template image and the actual image using a real-time pose position determination algorithm, and then calculates the pose parameters of the texture-less object using the minimum distance.

The voting-based methods: This type of method is mainly aimed at the presence of occlusion. The basis is that each part of the image can vote on the overall output. Tejani et al. [17] also developed a Hough forest-based approach for estimating 6DoF poses from RGB-D pictures. Li et al. [18] also proposed a multi-class CNN architecture with three novel features for accurate pose estimation. Kehl et al. [19] presented a local voting-based approach for performing reliable pose estimation in the conditions of occlusion and clutter. Furthermore, Wang et al. [20] presented the DenseFusion approach, which first extracts RGB features and geometric features using CNN for RGB and depth images, respectively, and then obtains fused pixel-level dense features, each of which can predict a pose, and finally obtains the final 6DoF pose by voting.

The regression-based methods: These methods focus on learning the connection between image features and 6DoF poses through CNNs. Regressing an object's 6DoF pose using deep learning is a typical strategy for target detection. There are two common approaches: one is to directly regress the 6DoF pose of the object end-to-end, and the other is to forecast the 3D key point's projection on the 2D image and then use the PnP method to recover the object's pose.

Among the end-to-end approaches for directly regressing an object's 6DoF pose, Xiang et al. [21] proposed the PoseCNN network for direct 6DoF object pose estimation, which estimates 3D translations by locating the distance between the object center and the predicted center, and then predicts 3D rotations by regressing to a quaternion representation. Kehl et al. [22] presented SSD6D, a new RGB data-based method for 6DoF pose estimation that extends the current SSD paradigm to encompass the entire 6DoF pose space. The methods mentioned above are either not capable of handling texture-less objects because they require quite enough texture surface information or have asymmetric shapes, or they are capable of handling texture-less objects but can’t estimate the 6DoF pose of texture-less objects in complex scenes, such as object occlusion. Different from the existing research, our method performs implicit reconstruction by acquiring templates for the full range of viewpoints of the texture-less object's 3D model, which effectively addresses the issue of estimating the pose of texture-less objects in an occluded scene. Also, to solve the multi-scale problem, the Feature Pyramid Network [23] was introduced.

III. METHOD AND MATERIAL

This section describes three main points: a) the generation method for the texture-less dataset; b) the implicit representation of single-view reconstruction; and c) the 6DoF pose estimation network architecture.

A. The generation method for the texture-less dataset

In order to meet the application requirements for sorting the texture-less bottles, a highly realistic and domain-randomized synthetic sample set was constructed for training, and some real samples were collected for testing.
Figure 1. Sample synthetic images for 6DoF pose estimation.

For the training dataset, we use Blender, a 3D computer graphics program that is free and open-source, to build a simple scene for rendering and generating data. The walls of the room are covered in a randomly selected material from the extremely realistic texture library, and light with a randomly selected color and strength is emitted from the room ceiling and from a randomly placed point light source in order to create a wide range of synthetic images, like Fig.1. Moreover, the COCO dataset [24] is used for background domain randomization, reducing the domain difference between real-world test images and synthetic images.

Figure 2. Sampling viewpoints by the Fibonacci grid.

The issue of how to sample as many 6DoF poses as possible on the object can be defined as how to arrange the sampling points evenly on the sphere. The Fibonacci grid was introduced as a new technique to sample synthetic samples for 6DoF pose estimation. Fig.2 shows the sampling viewpoints. Assuming that the radius of the sphere is 1 and that \( N \) sampling points are to be taken, the coordinates of the \( n^{th} \) point \((x_n,y_n,z_n)\) are given by the following expression:

\[
\begin{align*}
  x_n &= \frac{2^{n-1}}{N} - 1 \\
  y_n &= \frac{\sqrt{1 - z_n^2} \cdot \cos(2\pi n\phi)}{N} \\
  z_n &= \frac{\sqrt{1 - z_n^2} \cdot \sin(2\pi n\phi)}{N}
\end{align*}
\]

where \( \phi = (\sqrt{5} - 1) / 2 \approx 0.618 \).

For the test dataset, a texture-less bottle dataset for industrial sorting is created, which includes four bottles without discriminative color and texture. Unlike traditional computer vision tasks, the ArUco markers as a priori information to annotate the target were introduced since the 6DoF pose estimation task requires the acquisition of spatial information about the target. In this case, ArUco markers are square markers consisting of a wide black border that helps in fast detection in the image and an internal binary matrix for identifying the markers and providing error detection and correction. In fact, the ArUco marker is actually a code that can be used for camera pose estimation and camera correction.

Fig.3 shows how to use the ArUco markers to obtain the texture-less object’s pose. In the following steps, a frame in captured images can be manually annotated. Firstly, a better-annotated image and the texture-less object’s CAD model are inputted. Secondly, the CAD model is manually rotated and translated so that the model roughly matches the texture-less object in the image by PnP. Thirdly, the ICP algorithm is used to refine the poses to make them more accurate. Finally, the texture-less object’s 6DoF pose for that frame is obtained.

Figure 3. The texture-less bottle dataset annotation process.

Based on the transformation relationship among the camera, ArUco marker, and target object, combined with the texture-less object’s pose in one frame of the captured images, the texture-less object’s pose in the captured image set can be automatically obtained.

B. Implicit representation of single-view reconstruction

Due to characteristics such as the missing texture of texture-less objects, useful feature information cannot be extracted for 6DoF pose estimation. As a result, we present an approach for estimating the 6DoF pose of texture-less rigid objects by acquiring templates of all viewpoints of the texture-less object’s 3D model and then performing single-view reconstruction of the acquired RGB images. Inspired by the encoder-decoder, which is capable of self-learning to extract high-level abstract features of the input signal, an encoder-decoder network for single-view reconstruction was introduced.

Fig.4 depicts the encoder-decoder framework. There are two primary modules in the framework: the encoding process and the decoding process. The encoder maps the input sample \( x \in \mathbb{R}^D \) to the feature space \( z \) and then the decoder maps the abstract feature \( z \) back to the initial space to obtain the reconstructed sample \( x' \).

For single-view reconstruction, the network is trained to implicitly represent the input sample \( x \in \mathbb{R}^D \) as a low-dimensional feature space \( z \in \mathbb{R}^n \) with \( n \ll D \) [25]. The reconstructed sample \( x' \) can be represented by the following equation:

\[
x'(x) = \mathcal{F} \circ \mathcal{G}(x) = \mathcal{G}(x)
\]
Where $\mathcal{F}, \mathcal{G}$ are encoder and decoder respectively.

To enhance the learning of what is encoded by the implicit representation as well as to ignore the noise property, we apply the random augmentations $f_{\text{aug}}(\cdot)$ of the input image $x \in \mathbb{R}^2$ as follows:

$$x' = (\mathcal{F} \circ \mathcal{G} \circ f_{\text{aug}})(x) = \mathcal{G}(z) \quad (3)$$

The random augmentations include random translation, scale, and in-plane rotation. This representation is invariant to noise, so the reconstructed image can minimize the difference in domain between the real and synthetic image.

C. Network architecture

As depicted in Fig.4, the two-stage 6DoF pose estimation network structure is used to recover the pose of the texture-less object. The pipeline consists of a 2D detector, rotation estimation, translation estimation, and ICP refinement.

For the 2D detector, we use YOLOX [26], an experienced improvement of the YOLO series, where they changed the YOLO detector to operate in an anchor-free fashion and decoupled the prediction branch and combined it with the label assignment strategy SimOTA to achieve good results.

![Figure 4. Overview of the 6DoF pose estimation network structure.](image)

The network receives an original RGB image as input. And multi-scale feature maps are generated by the Feature Pyramid Network. These feature maps are then fed to YOLOX to obtain the 2D bounding box of the texture-less object. Based on the output of YOLOX, the original RGB image is cropped and fed into the rotation estimation network and the translation estimation network, respectively. In this case, the rotation estimation network obtains a codebook by encoding the cropped target object and then decodes it to obtain a 3D rotation estimate of the texture-less object. Depending on the scale of the bounding box, the translation estimation network can obtain a 3D translation estimate of the object. Finally, using the provided depth image, the obtained initial poses can be improved by using the ICP post-processing approach to create the texture-less object's final poses.

In particular, a cropped image with a 128x128 resolution serves as the rotation estimate network's input, which is convolved by a series of 5x5 sized convolution kernels to obtain a rotation encoding of dimension 128, and then reconstructed by deconvolution of a series of 5x5 sized convolution kernels to obtain a rotation estimation image of size 128x128 target object.

IV. EXPERIMENTAL STUDY

A. Metrics

In our experiments, our approach is evaluated using two metrics: the Average Distance of Model Points (ADD) [3] and the Visible Surface Discrepancy (VSD) [27].

**ADD metric.** For the ADD metric, the average distance between the set of model points converted from the estimated pose and the ground truth pose is calculated separately, and the estimated pose is considered correct when this distance is less than 10% of the model's diameter.

$$ADD = \frac{1}{m} \sum_{x \in \mathcal{M}} \| (Rx + t) - (\hat{R}x + \hat{t}) \| \quad (4)$$

**VSD metric.** The VSD metric is an ambiguity-invariant pose error function. The metric only computes the distance between the estimated and ground-truth visible object depth surfaces, which makes the metric invariant to ambiguities caused by occlusion and symmetries. Similar to the earlier work, we show the recall of the correct 6DoF object pose at $err_{vis} < 0.3$ with tolerance $r = 20\text{mm}$ and $> 10\%$ object visibility.

B. Implementation Details

In the experiments, the data enhancement is used for the training images. As proposed by Wu et al. [28], we adopt a bootstrapped pixel-wise L2 Loss. In addition, we employ the Adam optimizer, which has a learning rate of $2 \times 10^{-4}$, a batch size $= 64$ and 30000 iterations which take more than three hours on a single Nvidia GTX 2080Ti.

C. Results of the experiment

To compare ours with YOLO-6D[5] and PVNet, we implemented experiments on ADD metric and VSD metric. Table 1 and Table 2 respectively display the results.

**TABLE I. THE ACCURACY IN TERMS OF THE ADD METRIC.**

<table>
<thead>
<tr>
<th>Object</th>
<th>Methods</th>
<th>YOLO-6D</th>
<th>PVNet</th>
<th>Ours</th>
<th>Ours (ICP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>50.0</td>
<td>66.7</td>
<td>69.3</td>
<td>88.7</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>38.8</td>
<td>43.3</td>
<td>50.0</td>
<td>67.5</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>47.3</td>
<td>67.0</td>
<td>71.3</td>
<td>92.5</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>30.0</td>
<td>36.7</td>
<td>40.3</td>
<td>69.5</td>
</tr>
</tbody>
</table>

**TABLE II. THE ACCURACY IN TERMS OF THE VSD METRIC.**

<table>
<thead>
<tr>
<th>Object</th>
<th>Methods</th>
<th>YOLO-6D</th>
<th>PVNet</th>
<th>Ours</th>
<th>Ours (ICP)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>65.7</td>
<td>70.0</td>
<td>74.8</td>
<td>90.7</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>29.0</td>
<td>33.5</td>
<td>38.5</td>
<td>47.0</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>35.3</td>
<td>47.7</td>
<td>56.8</td>
<td>79.2</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>32.0</td>
<td>39.8</td>
<td>42.2</td>
<td>72.0</td>
</tr>
</tbody>
</table>

As illustrated in Tables 1 and 2, our approach performs better than the other two algorithms on both the ADD metric and the VSD metric for the texture-less bottle dataset.
In particular, the performance is further improved after combining the depth image of the texture-less object to perform ICP refinement. Despite the fact that our method produces superior results than the other two methods, on its own, there are still individual objects with a poor VSD metric.

The results of analyzing rotation and translation errors are displayed separately in Fig.5 and Fig.6. It is obvious that most of the views can estimate the rotation correctly. However, the Z-axis error for translation is relatively large due to the single view.

As described above, by sampling the template of the texture-less object, a single-view reconstructed image of the texture-less object can be obtained by the encoder-decoder network. Thus, the single-view reconstruction result can find the corresponding view according to the codebook, as shown in Fig.7.

![Single-view reconstruction results.](image)

**Figure 7.** Single-view reconstruction results.

Fig.8 shows the object pose results after ICP refinement. The iterative refinement of ICP has achieved satisfactory results.

![The object pose results after ICP iterative refinement.](image)

**Figure 8.** The object pose results after ICP iterative refinement.

### D. Ablation Study

In order to contrast how the results of the 6DoF pose estimate are affected by the CAD model and reconstruction model, we conducted an ablation study. The outcomes in Table 3 demonstrate that the CAD model outperforms the reconstructed model in both metrics.

**TABLE III. THE ACCURACY OF DIFFERENT TYPES OF OBJECT 2.**

<table>
<thead>
<tr>
<th>Metric</th>
<th>The type of model</th>
<th>CAD</th>
<th>Reconstruction</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADD</td>
<td>50.0</td>
<td>48.5</td>
<td></td>
</tr>
<tr>
<td>VSD</td>
<td>38.5</td>
<td>37.7</td>
<td></td>
</tr>
</tbody>
</table>

The reason why the CAD model gives better results than the reconstruction model could be the fact that the reconstruction model was obtained by using our own scanning equipment. During the scanning process, the interference of some external factors, such as the intensity of light and the accuracy of the scanning equipment, leads to poor scanning quality, so the accuracy of the reconstruction model is not enough.

### V. CONCLUSION

In industrial scenes, the lack of object texture information and the occlusion generated by stacking creates great challenges for 6DoF pose estimation. In order to conduct this study, two datasets were constructed: one was the training dataset created by using the 3D model of the texture-less object, and the other was the test dataset collected in a real-world scenario. And then a new framework, IRNet, for 6DoF pose estimation by the implicit representation of single-view reconstruction was proposed, which is capable of dealing with the occluded texture-less objects. According to the experimental outcomes, our approach outperforms the baseline methods for the occluded texture-less objects in complex industrial scenes. However, from the data of the experiments, it is clear that there exist individual objects with poor accuracy
in 6DoF pose estimation. Therefore, for future work, we will continue to optimize our network architecture as a way to improve the overall accuracy of texture-less objects as well as try to use them for industrial grasping and sorting.

ACKNOWLEDGEMENT
Our work is supported by multiple funds in China, including the Key Program of NSFC-Guangdong Joint Funds (U1801263), the Natural Science Foundation of Guangdong Province (2020A1515010890), the Industrial core and key technology plan of Zhusai City (ZH2204470219003HJL). Our work is also supported by Guangdong Provincial Key Laboratory of Cyber-Physical System (2020B1212060069).

REFERENCES
Hybrid Gait Recognition Method Based on Phase Trajectory

1st Pinjie Li  
Department of Automation  
Tsinghua University  
Beijing, China  
lpj18@tsinghua.org.cn

1st Tao Xue  
Department of Automation  
Tsinghua University  
Beijing, China  
xt16@tsinghua.org.cn

2nd Tao Zhang*  
Department of Automation  
Tsinghua University  
Beijing, China  
taozhang@tsinghua.edu.cn

3rd Huangang Wang  
Department of Automation  
Tsinghua University  
Beijing, China  
hgwang@tsinghua.edu.cn

4th Ming Zhang  
Department of information science and technology  
Shenzhen International Graduate School  
Tsinghua University  
Beijing, China  
m.zhang21@aston.ac.uk

Abstract—Gait recognition is critical to the activity monitoring, health management, assistance control of prostheses and exoskeletons, etc. This study aims to improve the gait classification performance on daily hybrid locomotions. We found the hip angle phase trajectories present significantly gait-dependent patterns, whereas the phase patterns show repeated limit cycles for periodic gaits while half-cycles or dots for aperiodic gaits. By converting the gait recognition issue into an image classification problem, we propose to use a convolution neural network (CNN) to learn the gait-dependent phase pattern images. Besides, to enhance the gait transition stability, we further integrate the prior state transition probability with category likelihood via dynamic Bayesian network. The proposed method has been experimented with 6 healthy subjects on standing, sitting, stand-to-sit, sit-to-stand, walking, running, stair ascending, and stair descending gaits. The overall leave-one-out cross-validation accuracy in continuous time is 96.15%.

Index Terms—Phase Trajectory, Convolution Neural Network, Dynamic Bayesian Network, Gait Recognition

I. Introduction

With the increased attention on healthcare, many wearable devices, i.e., smartphones, smartwatches, etc., have emerged to monitor human activities for exercise assessment, and health management [1]. Such monitor usually includes the collections on gait steps, movement time, trajectory, etc., but is insufficiently precise for gait analysis without gait mode recognition. Gait recognition is also critical to the assist-as-needed control of prostheses, orthoses, and exoskeletons as the required assistance varies due to different gait dynamics [2], [3]. So far, although various sensing technologies have been investigated on gait recognition, such as surface electromyogram (sEMG) sensing, optical tracking, inertia measurements, it remains challenging to obtain robust gait classification.

Machine learning methods have obtained promising results on gait estimations. Voicu et al. developed an activity recognition system based on built-in sensors in smartphones and used a multi-layer perceptron to learn the gait patterns. It obtained average 92% accuracy without stairs and 73% accuracy on the stairs gait [4]. Han et al. designed a decision tree structure with the neural network as the classifier, and the experimental results demonstrated 96.71% accuracy [5]. Kang et al. proposed a continuous gait phase-based Bayesian classifier to recognize walking, ramp, and stair gait. The author designed five individual phase-dependent Bayesian classifiers to get the class likelihood at current phase and fused it with prior state transition probability [6]. Neural networks specialize in discovering potential features beyond humans. As such, it might learn wrong patterns during training, in which the results seem good but over-fitted. In actual applications, it’s better to guide the network to learn correct patterns with human prior knowledge.

For our proposal, we target the recognition of level walking (LW), level running (LR), sitting, standing, sit-to-stand (SIT), stand-to-sit (STI), stair ascending (SA), stair descending (SD), and seamless gait transitions. To this end, we found the hip movements are highly correlated with gait modes, and its related phase trajectories present significantly different patterns for various gait modes. The phase maps show repeated limit cycles with gait-dependent shapes for periodic gait and half cycles or dots for aperiodic gaits. Therefore, it’s possible to use image pattern recognition techniques to learn these phase trajectory patterns and classify corresponding gaits.

The main contributions of this paper can be summarized as:

(1) With the hip angular features in phase space, we propose to use a CNN network to learn the gait-dependent patterns, in which the gait recognition issue is converted into an image classification problem. In this case, the gait segmentation and artificial feature extraction are no longer needed.
(2) The cross-validation results show comparable results to relevant studies, i.e., higher accuracy, lower latency, and simpler sensor. The overall recognition accuracy in continuous time domain is 96.15% increased by 8.07% with LDA methods.

II. Phase Analysis For Human Gait

In this study, we target the recognition of eight daily locomotions including standing, sitting, STI, ST, LW, LR, SA, and SD. The state transition map is shown in Fig. 1, where the eight gaits are divided into aperiodic mode and periodic mode. Aperiodic gait includes static stand and sit states as well as transient sit-to-stand and stand-to-sit states. In periodic gait, the lower-limb keeps changing the rotation directions periodically to finish the LW, LR, SA, or SD tasks. Note that the transition from stand to sit needs to pass through the stand-to-sit state, in which the static stand state to transient stand-to-sit state is bidirectional while the remaining process is a one-way process. It is the same case with the transition from sitting to stand. The state changes in periodic mode are all bidirectional except for SA and SD. Transitions between SA and SD are prohibited as this case rarely occurs in daily locomotion. Besides, the standing state is the necessary one for the switching between aperiodic and periodic gait.

The hip trajectories in aperiodic gaits are shown in Fig. 2. And the hip trajectories in periodic gaits are shown in Fig. 3. Although it can be seen that the angular trajectories present significant features of different gaits, it remains challenging to recognize the eight gaits from the angle features, for example, the key indicators, such as stance phase duty cycle and maximum flexion angle, are difficult to decide in real-time.

Note that the gait features also reflect on the hip angular velocity trajectories in addition to angle trajectories. The phase trajectory patterns of eight gaits are plotted in Fig. 4 and Fig. 5. The phase trajectories have significant contrasts that can be easily distinguished. The periodic gaits show close loops with various shapes while the aperiodic gaits present partial loops, dots, or lines. For static standing and sitting gaits, the trajectories converge to dots or lines on different angle locations. And they show opening upper semicircle and lower semicircle for STI and SIT, respectively. Even the phase maps present closed-loop for all periodic gaits, there are obvious differences in details, i.e., size, shape, location of embedded subloop, etc. The phase maps of both LW and LR have a closed loop with a subloop in the lower right section, but the LW seems faster than LR in the angle axis because the angular velocity in LR has a wider range. As for stair gait, the SA has a bigger angle range without the subloop while the SD gets a smaller angle range with a subloop in the lower-left section. Except for the significant patterns in phase space, another advantage of the phase trajectory method is that one no longer needs to care about the gait phase or cycle segmentation. We can restore the complete pattern with a relatively longer sliding
window since the overlapped trajectories have no impact on the phase patterns. Therefore, it becomes a feasible method to identify the eight gaits by comparing the 2D phase patterns, in which the gait classification problem is converted into a 2D pattern recognition problem.

![Phase trajectory patterns in standing, sitting, STI, and SIT gaits.](image1)

**Fig. 4.** Phase trajectories patterns in standing, sitting, STI, and SIT gaits.

![Phase trajectory patterns in LW, LR, SA, and SD gaits.](image2)

**Fig. 5.** Phase trajectory patterns in LW, LR, SA, and SD gaits.

### III. GAIT RECOGNITION ALGORITHM

#### A. Gait Mode Recognition Scheme

The overall locomotion recognition method is shown in Fig. 7. We use a simple CNN network is further designed to learn and classify these gait-dependent phase pattern images. The classification results from the network have some outliers due to the lack of sequential information. To this end, we add a gait transition filter at the end to fuse the gait likelihood from the network and prior gait transition probability. Meanwhile, we are able to prevent the prohibited gait transition by configuring the state transition matrix. The final gait estimation is obtained by maximizing the gait mode posterior.

#### B. CNN-based Phase Pattern Recognition

CNN has demonstrated its powerful learning capabilities, especially on image processing. In the proposed trajectory method, the gait features are converted into 2D image patterns. Therefore, we adopt CNN algorithm to recognize the eight gait patterns. The network architecture and detailed parameters are shown in Fig. 6 and Tab.(1).

![The architecture of CNN.](image3)

**Fig. 6.** The architecture of CNN. It includes convolution layers, batch normalization layers, pooling layers, reluLayers, fullconnection layers, and softmax layers.

The two-dimension signals $\theta_D$ and $\omega_z$ are integrated as one $100 \times 2$ image, which is regarded as the input of the network. The network first processes the image with a 2D convolution layer whose output channel is 64 and kernel size $40 \times 2 \times 1$. The output feature map flows into a batch norm layer, ReLU, and max-pooling layer in succession. The map is again processed with the second convolution layer, in which the channel number is set as 128 and the kernel size as $15 \times 1 \times 64$. The convolution layer is followed by a flatten layer, and it converts 2D image data into one-dimensional data. The converted feature vector is then fed into a full connection layer with eight nodes that presents eight gait modes. The values are further normalized with the softmax layer to generate probability distribution over the eight gaits.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Batch Norm</th>
<th>Activate</th>
<th>MaxPool</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageInput(100x2x1)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Conv2D(64, 40x2x1)</td>
<td>✓</td>
<td>ReLU</td>
<td>Max(2)</td>
</tr>
<tr>
<td>Conv2D(128, 15x1x64)</td>
<td>✓</td>
<td>ReLU</td>
<td>Max(2)</td>
</tr>
<tr>
<td>Flatten</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Linear(1024,8)</td>
<td>-</td>
<td>SoftMax</td>
<td>-</td>
</tr>
</tbody>
</table>

**TABLE I** Parameter settings of CNN network.

#### C. Gait Transition Filter

CNN classifies the gait from phase pattern images but without the consideration of sequential information, thus there might be a few outliers and unstable gait transitions. Actually, gait recognition is a sequential classification problem, where the current gait state depends on historical gait. Inspired by [6], we adopt dynamic Bayesian network to fill out the unstable points. Besides, we also integrate transition limitation as shown in Fig. 1 into this filter
by setting the corresponding element as zero in the state transition matrix.

The outputs from the softmax layer provide the current likelihood for each gait mode $P(x|C)$, in which $x$ denotes the input phase pattern and $C$ stands for the gait mode. The filtered gait mode at $k$ point is determined by maximizing the posterior probability as

$$C_k = \arg \max_C (P(C|x)). \tag{1}$$

According to Bayesian formula, the posterior probability is updated with the prior probability $P(C)$ and category likelihood as

$$P(C|x) = \frac{P(x|C)P(C)}{P(x)}. \tag{2}$$

The category likelihood can be obtained from the softmax layer of CNN. With the sequential property, we consider the gait as time-related random variables on the Markov assumption. Therefore, the prior probability can be derived by multiplying the posterior probability with the transition matrix $\Phi$.

$$P(C) = P(C|x)\Phi, \tag{3}$$

where $\phi_{i,j}$ denotes the transition probability from $C = i$ to $C = j$. The state transition matrix can be determined with train data as in [6]. To prohibit gait transition from $C = i$ to $C = j$, the transition probability element $\phi_{i,j}$ is set to zero. In this case, the prior probability and subsequent posterior probability on categories $j$ are dramatically decreased, which blocks the transition as expected.

IV. EXPERIMENTAL RESULTS

A. Data Preparation

The experiment collected data from 6 healthy subjects (one female, five males; 25±2 years old; 171±5 cm height; 65±10 Kg weight). The subjects were asked to perform LW, LR, SA, SD, standing, STI, sitting, SIT, were repeated twice with 1 minute interval. All the participants have given their consent before taking part in the study. The experimental protocol was approved by the Institute Review Board of Tsinghua University (No. 20200014). For convenience, the eight gait modes are labeled with numerical IDs as in Tab. II.

<table>
<thead>
<tr>
<th>Gait Mode</th>
<th>ID</th>
<th>Gait Mode</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing</td>
<td>1</td>
<td>Level Walking (LW)</td>
<td>5</td>
</tr>
<tr>
<td>Sitting</td>
<td>2</td>
<td>Level Running (LR)</td>
<td>6</td>
</tr>
<tr>
<td>Stand-to-Sit (ST1)</td>
<td>3</td>
<td>Stair Ascent (SA)</td>
<td>7</td>
</tr>
<tr>
<td>Sit-to-Stand (ST1)</td>
<td>4</td>
<td>Stair Descent (SD)</td>
<td>8</td>
</tr>
</tbody>
</table>

B. Experiment

In this study, recognition accuracy in the time domain, recognition delay, and hybrid gait recognition performance are investigated to evaluate the proposed algorithm performance. The accuracy is defined by the ratio between the correct identified sample points and the labeled points. The transition delay is calculated by the time interval between the recognized transition point and the labeled transition point. Regarding hybrid gait classification, we use the proposed method to recognize a series of continuous gaits, including LW, LR, SA, SD, Stand, Sit, STI, and SIT, to validate the robustness in daily complex gait. The classifier is trained with leave-one-out cross-validation, and the average and standard deviation results across 6 subjects are presented. To validate the superiority, we also compare the proposed method with the conventional pattern recognition method using a linear discriminant analysis (LDA) classifier. Six features of angle, including the first data, last data, max value, min value, mean value, and standard deviation in the sliding windows, and the same six features of angular velocity are chosen as the input for the LDA classifier. The features tend to yield better gait estimation performance as shown in [6], [10].
### TABLE III
Gait identification accuracy rate of six subjects.

<table>
<thead>
<tr>
<th></th>
<th>Stand</th>
<th>Sit</th>
<th>STI</th>
<th>ST</th>
<th>LW</th>
<th>LR</th>
<th>SA</th>
<th>SD</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>S#1</td>
<td>96.20</td>
<td>95.12</td>
<td>101.00</td>
<td>94.15</td>
<td>97.48</td>
<td>95.36</td>
<td>96.13</td>
<td>92.38</td>
<td>96.71</td>
</tr>
<tr>
<td>S#2</td>
<td>92.44</td>
<td>100.00</td>
<td>83.34</td>
<td>87.65</td>
<td>98.10</td>
<td>98.73</td>
<td>90.66</td>
<td>91.40</td>
<td>95.34</td>
</tr>
<tr>
<td>S#3</td>
<td>95.52</td>
<td>96.77</td>
<td>72.22</td>
<td>100.00</td>
<td>98.46</td>
<td>98.68</td>
<td>96.23</td>
<td>94.91</td>
<td>96.76</td>
</tr>
<tr>
<td>S#4</td>
<td>92.56</td>
<td>86.30</td>
<td>100.00</td>
<td>92.31</td>
<td>93.35</td>
<td>95.53</td>
<td>96.25</td>
<td>97.14</td>
<td>94.48</td>
</tr>
<tr>
<td>S#5</td>
<td>93.47</td>
<td>94.59</td>
<td>98.89</td>
<td>86.01</td>
<td>99.06</td>
<td>98.39</td>
<td>92.61</td>
<td>97.73</td>
<td>95.76</td>
</tr>
<tr>
<td>S#6</td>
<td>97.00</td>
<td>97.50</td>
<td>90.00</td>
<td>91.74</td>
<td>96.57</td>
<td>100.00</td>
<td>100.00</td>
<td>99.72</td>
<td>97.89</td>
</tr>
</tbody>
</table>

### C. Results

The results are shown in Tab. III, where S# 1-6 denotes the 6 participants. The results of aperiodic gait vary from 72.22% to 100%, but they show high consistency for periodic gait. The worst case is 72.22% that occurs in STI transient gait with S#3, and the best case 100% happens in both aperiodic and periodic gait. The averaged gait classification confusion matrix across 6 subjects is shown in Fig. 8. The correct recognized gait rate ranges from 90.52% to 98.52%. STI is the worst case with 90.52% accuracy, and it could be mistakenly identified as adjacent standing and sitting gait. It’s because transient gait has short spans and even slight misalignment will result in a sharp decline in accuracy. The same analysis also applies to STI gait. Standing gait can be wrongly identified as any other gait except for Sit, and Sit can be only mistakenly identified as transient STI and STI gait. Regarding periodic gait, the highest classification accuracy is LR with 98.52%, followed by LW with 97.17%, SD with 96.22%, and SA with 94.69%. The proposed method has less confidence on SA recognition, in which the SA is most likely considered as LW with a 3.67% rate. Note that all the periodic gait can be wrongly classified as stand. The true LW gait can be regarded as LR, SA, and SD, and true LR can be considered as LW, SA and SD won’t be misclassified for each other, but they can be identified as LW and LR.

In Fig. 9, the classification performance of the proposed filtered CNN (FCNN) method, CNN method, and LDA methods are compared. Compared with LDA, the FCNN shows significant accuracy improvements from 2.77% to 15.12% on each gait, especially on SA, SD with 15.12% and 14.97% promotion. For overall accuracy, the FCNN method also shows comparable results to LDA, in which the total accuracy is increased by 8.07% from 88.97% to 96.15%. Besides, it can be obtained that standard CNN presents better accuracy than the LDA method, and the proposed FCNN method with added gait transition filter further improves the recognition performance.

Note that the presented recognition accuracy here is time-based, which might be more conservative when compared with commonly used event-based statistics. In the event-based case, people usually take the periodic gait as a discrete variable and assume the gait mode is constant during one complete gait cycle. However, this method excludes transient gait and ignores the gait changing over one gait cycle, in which the recognition

### Fig. 8.
The confusion matrix of gait classification accuracy (%), whereas the horizontal axis denotes recognized gait mode with proposed algorithm and the vertical axis represents the ground truth. The recognition results are calculated using the combination of the validation data across six subjects.

### Fig. 9.
Gait mode classification performance with improved FCNN, CNN, and LDA methods. The presented data is the average recognition accuracy (%) with standard deviation across six subjects. The results include the individual gait accuracy and total accuracy. The number above the histogram denotes the accuracy improvement rate of FCNN compared with the LDA method.
results seem higher than the actual case. By contrast, the proposed classification algorithm is executed on each coming sample in the whole gait cycle. In this scheme, the identification error is inevitable if we include the transient gait. It is because gait transition is not instant and the transient gait between two consecutive steady gaits has both features, which is difficult to denote as a specific state. The results in Fig. 8 also support the analysis, whereas the misclassified gait only occurs when the two gaits have transition connections. To make the results convincing, we compare the recognition performance of FCNN, CNN, and LDA in the same statistical method in Fig. 9. As expected, it demonstrates better performance with proposed phase trajectory-based methods and further enhanced accuracy with an additional gait transition filter.

V. DISCUSSIONS

The experiments demonstrate that it is a feasible method to recognize the gait mode with phase trajectory pattern classification. One reason for choosing the phase trajectory as the input because the hip angle and hip angular velocity can be easily obtained from a single IMU sensor. Another reason is that the phase trajectory shows a stable limit cycle in a steady periodic gait. In this case, the phase trajectory patterns remain constant as long as we keep the historical data over one gait cycle. Moreover, these phase patterns in 2D space are highly gait-dependent, which makes it possible to use image classification techniques. We take each phase pattern as a special character and convert the gait recognition issue into a character classification problem. We choose the CNN network to learn the gait patterns since it has significant potential for image classification. The CNN method classifies the gait through 2D pattern recognition without consideration of the time sequence of gait, in which the results might include a few jump points in steady gait. To this end, the prior probability from the state transient matrix and current likelihood obtained by the network are fused through Bayesian filter. Our method shows comparable results. And our method is not gait event-based but phase pattern-based, thus gait segmentation is no longer necessary. Besides, it’s difficult for a conventional method to design one general rule manually to classify the diverse gaits of multiple subjects. But, the CNN network can distinguish the gait differences on hip kinematics from the view of image patterns. Moreover, this method can be expanded to classify any other gaits, i.e., slop walking, slop running, etc., as long as we train the CNN network with corresponding gait data.

Note that the presented recognition accuracy in this paper is more conservative. Some research claim 100% recognition accuracy and I assume the accuracy means steady gait recognition without transient gait. To enhance practicability, we asked the subject to perform a series of daily gait including periodic gait, aperiodic gait as well as various gait transitions. We calculate the overall recognition results but not a segmented interval, in which the transient gait is hard to define but we labeled it as a certain gait according to the context. It will introduce extra errors with our statistical method.

One limitation of our method is the demand for computing power even though the CNN structure is simple with just two convolution layers and one full connection layer. Another issue is the data, we need to collect the gait data to train the network, and the diversity data will enhance the classification generalization.

VI. CONCLUSIONS

This study proposes a hybrid gait recognition method based on phase trajectory by transforming the gait recognition problem into an image classification problem, using convolutional neural networks (CNN) and Bayesian filters to classify gait-related patterns. The proposed method has been experimented with 6 healthy subjects on standing, sitting, stand-to-sit, sit-to-stand, walking, running, stair ascending, and stair descending gaits. It showed comparable results to related studies. More importantly, this method can be easily implemented on smart wearable devices, prostheses, orthoses, exoskeletons, etc.

References

Adaptive Fixed-time Fuzzy Tracking Control of Uncertain Robot

Beibei Zhou, Chengzhi Zhu, and Yiming Jiang*

Abstract—We propose a novel fixed-time fuzzy control strategy to achieve the trajectory tracking of the robot systems with uncertainties in this article. The tracking errors of the system can be convergent to a small area around zero with fixed-time convergence rate, combining with the fuzzy logic system and the backstepping technique. Besides, the boundedness of all the closed-loop system signals can be proved under the proposed controller. The superiority performance of the proposed control strategy has been shown by the simulation results.

Index Terms—fixed-time control, fuzzy logic system, robot systems

I. INTRODUCTION

The researches on tracking control with the desired performance have attracted wide attention in the robotics community. In the recent decades, many model-based control algorithms based on accurate model information have been extensively discussed [1], [2]. However, the model information cannot be accurately obtained in most situations because of the time-varying uncertainties in the robot systems, which hinders the development of such control schemes.

The backstepping technique, as a regression method, is popular in the nonlinear control community due to its simple construction and ease to implement [3]–[8]. Thus, it is often used in combination with the universal approximation methods, such as neural network (NN) [9]–[12] and fuzzy logic system (FLS) [13]–[16], handling the control problems of the uncertain systems. In [9], a modified NN has been proposed to compensate friction effectively. In [12], a novel adaptive NN control scheme has been proposed to optimize the system convergence time. In [13], a novel decentralized fuzzy logic controller (FLC) has been developed to approximate the unknown terms for the double arm robot and the manipulated object. In [16], an adaptive singularity-free fuzzy control strategy has been proposed to address the tracking problem for the uncertain n-link robotic system.

In the existing researches, the system convergence time has been rarely involved. In the recent decades, the definition of the finite-time convergence [17]–[19] was proposed in amounts of control design. In [17], the finite-time convergence of the system signals can be achieved with the designed parameter identification and control scheme for the uncertain robots. In [18], an approximation-free finite-time controller was presented for the uncertain quadrotor UAVs. However, the convergence time is always related to the initial system states in the existing finite-time control algorithms, which is a common drawback of such control schemes. The fixed-time control schemes [20]–[22] have been proposed to overcome the imperfection. In [21], a novel observer-based fixed-time controller has been presented for uncertain robots to obtain the superior tracking performance. In [22], a novel adaptive fixed-time controller based on the parameter identification has been proposed for the uncertain robots to guarantee the output constraints.

Inspired by these researches, a novel fixed-time fuzzy controller is designed for the uncertain robot systems in this article. We list the major contributions of this article as follows:

1) All the system signals can achieve semiglobal uniformly ultimately boundedness (SGUUB) with the proposed fixed-time controller.
2) Each tracking error signal can converge to the compact set around zero in the predefined convergence time regardless of the initial states.

II. PROBLEM FORMULATION AND PRELIMINARIES

A. Problem Formulation

Consider a general n-link robot system as follows:

$$\dot{M}(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = \tau$$

where $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$ denote the joint position, velocity, and acceleration respectively. $\tau \in \mathbb{R}^n$ can be shown as the input torque. $M(q), C(q, \dot{q}) \in \mathbb{R}^{n \times n}$ represent the unknown inertia matrix, Coriolis and centripetal torque matrix, respectively. $G(q) \in \mathbb{R}^n$ denotes the unknown gravity vector. And the controller design strategy can be shown as Fig. 1.

B. Preliminaries

Assumption 1: Both the derivative of the desired trajectory $q_d$ and itself are continuous and bounded. In addition, the inertia matrix $M(q)$ in (1) is symmetric and positive definite.

Property 1 [23]: For the expression $\dot{M}(q) - 2C(q, \dot{q})$ in (1), the following property always exists:

$$\dot{q}^T \left( \dot{M}(q) - 2C(q, \dot{q}) \right) \dot{q} = 0, \forall q \in \mathbb{R}^n.$$
Fig. 1. Block diagram of the controlled robot system

**Lemma 1** [22]: When the general Lyapunov function $V(\xi)$ satisfies that $\dot{V}(\xi) \leq -h_1 V^{\alpha_1}(\xi) - h_2 V^{\alpha_2}(\xi) + \rho$ for the system $\xi(t) = f(\xi)$, the system is practical fixed-time convergent with the following convergence region:

$$\Omega_1 = \left\{ \xi \mid V \leq \min \left( \frac{\rho}{h_1(1-\varsigma)} \right)^{1/\alpha_1}, \frac{\rho}{h_2(1-\varsigma)} \right\}$$

where $h_1$, $h_2$, $\rho$ are all positive constants, and $\varsigma$, $\alpha_1$ and $\alpha_2$ satisfy that $\varsigma, \alpha_1 \in (0,1)$ and $\alpha_2 \in (1, +\infty)$, respectively. The system convergence time is: $T \leq T_{max} = \frac{1}{h_1(1-\varsigma)} + \frac{1}{\kappa_2(\alpha_2-1)}$.

**Lemma 2** [24]: The following expressions always exist for any $\beta_i \in R$ and $\eta \in R^+$:

$$\sum_{i=1}^{n} |\beta_i|^\eta \geq \left( \frac{1}{\sum_{i=1}^{n} |\beta_i|} \right)^\eta, 0 < \eta \leq 1$$

$$\sum_{i=1}^{n} |\beta_i|^\eta \geq n^{1-\eta} \left( \sum_{i=1}^{n} |\beta_i| \right)^\eta, \eta > 1$$

**Remark 1** [25]: A newly defined function is shown as follows:

$$\text{sign}^v(\delta) = |\delta|^v \text{sign}(\delta)$$

where $\delta \in R$, $v \in R^+$, and $\text{sign}(\cdot)$ represents the signum function.

**Fuzzy Logic System** [13]: The fuzzy If-then rules in the knowledge base are expressed as:

R1: if $x_1$ is $D_1^L$, $D_2^L$, $x_3$ is $D_3^L$

Then $\Psi$ is $E^L$, $l = 1, \ldots, N$.

The terms $x = [x_1, \ldots, x_N]^T \in R^n$ and $\Psi$ respectively represent the input and output of the FLS. The fuzzy sets and their corresponding membership functions can be denoted as $D^L, E^L$ and $\mu_{D^L}(x_i), \mu_{E^L}(\Psi)$, respectively. The total amount of the fuzzy rules is $N$.

Then the FLS can be shown as:

$$\Psi(x) = \frac{\sum_{i=1}^{N} \Psi_i \prod_{i=1}^{n} \mu_{D^L}(x_i)}{\sum_{i=1}^{N} \prod_{i=1}^{n} \mu_{D^L}(x_i)}$$

where $\Psi_i = \max_{\Psi \in R} \mu_{E^L}(\Psi)$. Define the fuzzy basis function as:

$$\Phi_i = \frac{\prod_{i=1}^{n} \mu_{D^L}(x_i)}{\sum_{i=1}^{N} \prod_{i=1}^{n} \mu_{D^L}(x_i)}$$

then the FLS (5) can be represented as:

$$\Psi(x) = \Theta^T \Phi(x)$$

**Lemma 3** [13]: For an arbitrary continuous function $F(x), x \in \Xi$ where $\Xi$ is a compact set, the following expression holds for any $\epsilon \in R^+$:

$$\sup_{x \in \Xi} |F(x) - \Theta^T \Phi(x)| \leq \epsilon$$

where $\Theta^T \Phi(x)$ is the FLS defined in (7).

## III. CONTROL DESIGN

An adaptive fixed-time tracking control strategy will be proposed combining the FLS and backstepping technique in this section. Firstly, we define the tracking error signals as follows:

$$s_1 = q - q_d$$

$$s_2 = \dot{q} - \dot{q}_d$$

where $s_1 = [s_{11}, s_{12}, \ldots, s_{1n}]^T$, $s_2 = [s_{21}, s_{22}, \ldots, s_{2n}]^T \in R^n$ represent the robot position and velocity tracking error vectors, respectively. The virtual controller vector can be represented as $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n] \in R^n$ and will be defined in (13).

Differentiating (9) with respect to time, we have:

$$s_{ij} = s_{2j} + \alpha_j - \dot{q}_{dj} \quad j = 1, 2, \ldots, n$$

We define a candidate Lyapunov function as follows:

$$L_1 = \frac{1}{2} \sum_{j=1}^{n} s_{1j}^2$$

From (10) and (11), we can get:

$$\dot{L}_1 = \sum_{j=1}^{n} s_{1j} \dot{s}_{1j} = \sum_{j=1}^{n} s_{1j}(s_{2j} + \alpha_j - \dot{q}_{dj})$$

Then, the virtual controller $\alpha_j$ is constructed as:

$$\alpha_j = \dot{q}_{dj} - \kappa_{1j}s_{1j} - \kappa_{12} s_{1j}^5 - \kappa_{13}s_{1j}^{1/3}(s_{1j})$$

where the definition of $\text{sign}^v(\cdot)$ can be found in Remark 1 and $\kappa_{1j}, \kappa_{12}, \kappa_{13}$ are all designed as proper positive constants.

From (12) and (13), we can get:

$$\dot{L}_1 = \sum_{j=1}^{n} s_{1j}(s_{2j} - \kappa_{1j}s_{1j} - \kappa_{12}s_{1j}^5 - \kappa_{13}s_{1j}^{1/3}(s_{1j}))$$

$$= \sum_{j=1}^{n} (s_{1j}s_{2j} - \kappa_{1j}s_{1j}^2 - \kappa_{12}s_{1j}^6 - \kappa_{13}|s_{1j}|^{2/3})$$

352
Then the candidate Lyapunov function is designed as:

$$L_2 = L_1 + \frac{1}{2} s_2^T M s_2$$

Taking the derivative of (15), we have:

$$\dot{L}_2 = \dot{L}_1 + \frac{1}{2} s_2^T M s_2 + s_2^T (\tau - C\dot{q} - G)$$

From (1) we can get:

$$\dot{q} = M^{-1}(\tau - C\dot{q} - G)$$

where $M$, $C$, $G$ are the abbreviation of the corresponding terms in (1) respectively.

Substituting (9) and (17) into (16) combined with Property 1, we have:

$$\dot{L}_2 = \dot{L}_1 + \frac{1}{2} s_2^T M s_2 + s_2^T (\tau - C\dot{q} - G - M\dot{\alpha})$$

where $S = [q^T, \dot{q}^T, \alpha^T, \dot{\alpha}^T]^T \in R^{4n}$. The unknown vector $F(S) = [f_1(S), f_2(S), ..., f_n(S)]^T \in R^n$ is designed as:

$$F(S) = -M\dot{\alpha} - C\dot{q} - G$$

The FLSs in II-B are employed to deal with the approximation of the unknown functions, then $f_j(S)$ and its estimation $\hat{f}_j(S)$ can be shown as:

$$f_j(S) = \Theta_j^T \Phi_j(S) + \epsilon_j, |\epsilon_j| \leq \epsilon_j$$

$$\hat{f}_j(S) = \hat{\Theta}_j^T \Phi_j(S)$$

where $\Theta_j^T \in R^{n_j}$ and $\hat{\Theta}_j^T \in R^{n_j}$ can be regarded as the optimal weight vector and its estimation, respectively. The basic function vector is $\Phi_j(S) = [\Phi_{1j}(S), ..., \Phi_{\chi_j}(S)]^T$ with $\chi_j$ being the number of fuzzy rules, and $\epsilon_j$ is the approximation error bounded by $|\epsilon_j| \leq \epsilon_j$. For convenience, we abbreviate $f_j(S), \hat{f}_j(S), \Phi_j(S)$ to $f_j, \hat{f}_j, \Phi_j$ respectively in the later design process.

The fuzzy fixed-time controller is designed as:

$$\tau_j = -s_{1j} - (\kappa_{I1j} + \frac{1}{2})s_{2j} - \hat{f}_j - \kappa_{I5j} s_{2j} - \kappa_{I6j} s_{1}^{1/3} (s_{2j})$$

where $\kappa_{I1j}, \kappa_{I5j}, \kappa_{I6j}$ are all designed as proper positive constants.

The adaptive law in the fuzzy control scheme is designed as:

$$\dot{\hat{\Theta}}_j = Q_j \left( s_{2j} \Phi_j - w_j \hat{\Theta}_j \right)$$

where $Q_j$ and $w_j$ are the positive definitive diagonal matrix and the positive parameter, respectively.

**Theorem 1:** When the proposed fuzzy controller in (21) is applied to the uncertain robot in (1), the fixed-time convergence can be guaranteed for the tracking errors regardless of the initial system states. Furthermore, each closed-loop signal of the system can achieve SGUUB.

**Proof:** Consider the global Lyapunov function as follows:

$$L = L_2 + \frac{1}{2} \sum_{j=1}^{n} \hat{\Theta}_j^T Q_j^{-1} \hat{\Theta}_j$$

where $\hat{\Theta}_j = \Theta_j - \Theta_j$ and $\Theta_j$ is the estimation error of $\Theta_j$.

Take the derivative of $L$ combined with (14) and (18), we can get:

$$\dot{L} = \dot{L}_1 + s_2^T (\tau + F(S)) - \sum_{j=1}^{n} \hat{\Theta}_j^T Q_j^{-1} \hat{\Theta}_j$$

$$= \sum_{j=1}^{n} (s_{1j} s_{2j} - \hat{\Theta}_j^T Q_j^{-1} \hat{\Theta}_j)$$

$$- \sum_{j=1}^{n} (\kappa_{I1j} s_{1j}^2 + \kappa_{I2j} s_{1j}^6 + \kappa_{I3j} |s_{1j}|)$$

$$\leq \sum_{j=1}^{n} (s_{2j} \tau_j + s_{2j} f_j)$$

Substituting (21) and (22) into (24), we have:

$$\dot{L} = - \sum_{j=1}^{n} (\kappa_{I1j} s_{1j}^2 + \kappa_{I2j} s_{1j}^6 + \kappa_{I3j} |s_{1j}|)$$

$$- \sum_{j=1}^{n} (\kappa_{I4j} s_{2j}^2 + \frac{1}{2} s_{2j}^2 + s_{2j} \hat{f}_j + \kappa_{I5j} s_{2j}^6)$$

$$\leq \sum_{j=1}^{n} (s_{2j} \epsilon_j + \frac{1}{2} s_{2j}^2 + w_j \hat{\Theta}_j^T \hat{\Theta}_j)$$

Substituting (20) into (25), we have:

$$\dot{L} = - \sum_{j=1}^{n} (\kappa_{I1j} s_{1j}^2 + \kappa_{I2j} s_{1j}^6 + \kappa_{I3j} |s_{1j}|)$$

$$- \sum_{j=1}^{n} (\kappa_{I4j} s_{2j}^2 + \kappa_{I5j} s_{2j}^6 + \kappa_{I6j} |s_{2j}|)$$

$$\leq \sum_{j=1}^{n} (s_{2j} \epsilon_j - \frac{1}{2} s_{2j}^2 + w_j \hat{\Theta}_j^T \hat{\Theta}_j)$$

In the light of the Young’s inequality, we can obtain the following results:

$$s_{2j} \epsilon_j \leq \frac{1}{2} s_{2j}^2 + \frac{1}{2} s_{2j}^2$$

$$w_j \hat{\Theta}_j^T \hat{\Theta}_j = -w_j \left\| \hat{\Theta}_j \right\|^2 + w_j \hat{\Theta}_j^T \hat{\Theta}_j$$

$$\leq \frac{1}{2} w_j \left\| \hat{\Theta}_j \right\|^2 + \frac{1}{2} w_j \left\| \hat{\Theta}_j \right\|^2$$
Substituting (27) and (28) into (26), we can get:
\[
\dot{L} = -\sum_{j=1}^{n}(\kappa I_{1j}s_{1j}^2 + \kappa I_{4j}s_{2j}^2) \\
- \sum_{j=1}^{n}(\kappa I_{2j}s_{1j}^6 + \kappa I_{5j}s_{2j}^6) \\
- \sum_{j=1}^{n}(\kappa I_{3j}|s_{1j}|^4 + \kappa I_{6j}|s_{2j}|^4) \\
+ \sum_{j=1}^{n}\left(\frac{1}{2}w_j^2 + \frac{1}{2}w_j\|\Theta_j^*\|^2 - \frac{1}{2}w_j\|\dot{\Theta}_j\|^2\right)
\]  
(29)

Furthermore, (29) becomes:
\[
\dot{L} \leq -\sum_{j=1}^{n}(\kappa I_{1j}s_{1j}^2 + \kappa I_{4j}s_{2j}^2 - \frac{1}{2}\epsilon_j^2) \\
+ \sum_{j=1}^{n}\left(\frac{1}{2}w_j\|\Theta_j^*\|^2 - \frac{1}{2}w_j\|\dot{\Theta}_j\|^2\right)
\]  
(30)

\[
\leq -\sum_{j=1}^{n}(\kappa I_{1j}\left(\frac{1}{2}s_{1j}^2\right) + \frac{2\kappa I_{4j}}{2\lambda_{\max}(M)}s_{2j}^2) \\
- \sum_{j=1}^{n}\frac{w_j}{\lambda_{\max}(Q_j^{-1})}\frac{1}{2}\lambda_{\max}(Q_j^{-1})\|\dot{\Theta}_j\|^2 \\
+ \sum_{j=1}^{n}\left(\frac{1}{2}w_j^2 + \frac{1}{2}w_j\|\Theta_j^*\|^2\right) \\
\leq -\min(2\kappa I_{1j})L_1 - \min\left(\frac{2\kappa I_{4j}}{\lambda_{\max}(M)}\right)\frac{1}{2}w_j^TMs_2 \\
- \min\left(\frac{w_j}{\lambda_{\max}(Q_j^{-1})}\right)\sum_{j=1}^{n}\left(\frac{1}{2}\Theta_j^TQ_j^{-1}\dot{\Theta}_j\right) + \epsilon_1 \\
\leq -a_1L + \epsilon_1
\]

where \( a_1 = \min\left(2\kappa I_{1j}, \frac{2\kappa I_{4j}}{\lambda_{\max}(M)}, \frac{w_j}{\lambda_{\max}(Q_j^{-1})}\right),\) \( \epsilon_1 = \sum_{j=1}^{n}\frac{1}{2}\epsilon_j^2 + \sum_{j=1}^{n}\frac{1}{2}w_j\|\Theta_j^*\|^2 \) is a bounded positive constant. And \( \lambda_{\max}(\cdot) \) is used to represent the maximum eigenvalue of the matrix.

Multiply the equation (30) by \( e^{a_1t} \), and integrating over \([0, t]\), we can get:
\[
L \leq \left[L(0) - \frac{\epsilon_1}{a_1}\right]e^{-a_1t} + \frac{\epsilon_1}{a_1} \leq L(0) + \frac{\epsilon_1}{a_1}
\]  
(31)

**Remark 2:** It’s obvious from (31) that the Lyapunov function \( L \) is bounded. Then according to the definitions of \( L_1, L_2, L \) in (11), (15), and (23), it can be concluded that \( s_{1j}, s_{2j}, \) and \( \dot{\Theta}_j \) are all bounded. Then we can prove the virtual controller \( \alpha_j \) is bounded by its expression in (15) combined with Assumption 1. Then \( q \) and \( \dot{q} \) can be both concluded bounded from (9). Furthermore, the boundedness of the controller \( r_j \) in (21) can be obtained from the boundedness of \( \Theta_j \) and \( \Phi_j(S) \). So far, all the closed-loop signals of the system have been proved to achieve SGUB.
where \( a_2 = \frac{1}{\delta_2} \min \left( \frac{8\kappa_{I2j}}{\lambda_{\max}(M)}, \frac{8\kappa_{I7j}}{\lambda_{\max}(Q_{1j}^{-1})} \right) \) and

\[
a_3 = \min \left( \kappa_{I3j} (\frac{1}{2})^2, \ldots \right.
\]

From (35) and Lemma 1, the fixed-time convergence of the system can be guaranteed regardless of the initial system states with the convergence residual region of

\[
\Omega_z = \left\{ x \mid |L| \leq \min \left( \left( \frac{\varepsilon_2}{a_3 (1 - t_1)} \right)^{\frac{2}{3}}, \left( \frac{\varepsilon_2}{a_2 (1 - t_1)} \right)^{\frac{2}{3}} \right) \right\},
\]

(36)

where \( \varepsilon_1 \) is a constant which satisfies \( \varepsilon_1 \in (0, 1) \).

In addition, the convergence time can be obtained:

\[
T_z = \frac{3}{a_{31}} + \frac{1}{2a_{21}}
\]

(37)

So far, the proof of Theorem 1 has been finished.

**IV. SIMULATION STUDY**

An uncertain 2-DOF robot system is used to validate the feasibility of the proposed fixed-time fuzzy controller in (21) in this section.

When \( n = 2 \), the robot model parameters in (1) are represented as follows:

\[
M(q) = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix},
\]

\[
C(q, \dot{q}) = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix},
\]

\[
G(q) = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix}
\]

(38)

where the terms in (38) have the same definition as \( \mathcal{M}(\xi), \mathcal{C}(\xi, \dot{\xi}), \mathcal{G}(\xi) \) in the simulation examples [16] with \( m_1 = 2 \text{kg}, l_1 = l_2 = 0.5 \text{m}, I_1 = I_2 = 0.125 \text{kgm}^2, g = 9.8 \text{m/s}^2 \).

The reference trajectory \( q_d \) is designed as \( q_d = \frac{\pi/3 + 0.5 \cos(2t) - 0.5}{5} \), \( \pi/5 + 0.6 \sin(2t) \). Four different initial conditions are set as \( [0.3, -0.3]^T, [0.7, -0.7]^T, [1.1, -1.1]^T, [1.5, -1.5]^T \) to approve the fixed-time convergence. The control gains are selected as \( \kappa_{I11} = \kappa_{I21} = \kappa_{I31} = 4, \kappa_{I12} = \kappa_{I22} = \kappa_{I32} = 1.8, \kappa_{I41} = \kappa_{I51} = \kappa_{I61} = 10, \kappa_{I42} = \kappa_{I52} = \kappa_{I62} = 5 \). And the gains of the fuzzy adaptive law in (22) are designed as \( Q_1 = \text{diag} [1.2], Q_2 = \text{diag} [1.8] \) and \( w_1 = 0.5, w_2 = 0.1 \).

Figs. 2-4 are the simulation results. Fig. 2 has shown the fast tracking performance of the selected uncertain robot with different initial conditions. Figs. 3-4 have shown the control input \( \tau_j \) and the norm of fuzzy weight estimation \( \Theta_j \). The boundedness of the closed-loop system signals can be seen from the above results. In addition, the convergence time of \( s_{1j} \) and \( \Theta_j \) is nearly the same when the initial errors increase. Then, the superiority of the proposed fixed-time controller can be seen in terms of the convergence time.

**V. CONCLUSION**

A novel fixed-time fuzzy control algorithm is discussed for uncertain robots in this article, where the FLSs are applied to handle the system uncertainties. With the proposed controller, all the system signals can achieve SGUUB and the convergence time can be predefined regardless of the initial states. The superior simulation results on a 2-DOF robot have
clarified the effectiveness of our work.

REFERENCES


Detection and Removal of EOG Artifact from EEG Signal using Fuzzy Logic and Wavelet Transform

1Said AGOUNAD, 1Hafida IDRISSI AZAMI, 1Mustapha MOUFASSIH, 1Ousama TARAHI, 1Soukaina HAMOU

1Laboratory of Metrology and Information Processing, Physics Department, Faculty of Sciences, Ibn Zohr University Agadir Morocco
Email: s.agounad@uiz.ac.ma, h.azami@uiz.ac.ma, mustapha.moufassih@edu.uiz.ac.ma, ousama.tarahi@edu.uiz.ac.ma, soukaina.hamou96@gmail.com

Abstract—The EEG signals (electroencephalogram) are one of the most used biosignals in diagnosis of neurological disorders and technology like brain computer interface (BCI). The main problem in the use of EEG signals is that are often contaminated by physiological and non-physiological artifacts. EOG (electrooculogram) artifact is one of the first origins of artifacts in EEG recordings. It severely affects the EEG signal and modifies the shape of the neural activity that drive the BCI system affecting then its accuracy and performance. This paper presents an original algorithm to detect and remove EOG artifacts from EEG signals. In the proposed method, the statistical parameters extracted from the signal are used as the inputs of a fuzzy inference system to take a decision if an epoch of EEG is contaminated by the EOG or not. Thereafter, the level-5 stationary wavelet transform (SWT) is applied on the detected contaminated epochs to remove the artifacts without bringing much distortion to the neural activity. The performance of the developed algorithm is demonstrated in two scenarios: semi-simulated artifactual EEG data and fully real artifactual EEG data. The presented results show that our algorithm outperforms some existing algorithms.

Keywords—Brain computer interface, Fuzzy logic, biomedical signal processing, artifact handling; wavelet transform.

I. INTRODUCTION

The EEG signal is measured using a non-invasive technique. It is used in diagnosis and research since it is characterized by high temporal resolution, portability, and lower cost. In diagnosis field, EEG signal is used to study sleep disorder [1], neurological diseases [2,3], effects of drug [4], etc. In research field, EEG signal is used in BCI systems [5], neuro feedback applications [6], sports sciences [7], etc. The EEG signal composes of different rhythmic activities: δ, θ, α, β, and γ, which are characterized by the following frequency band, respectively: (0.5 to 4 Hz), (4 to 8 Hz), (8 to 12 Hz), (13 to 31 Hz) and (31 to 60 Hz). The amplitude of the EEG activity is usually varied between 10 and 50 μV which is quite weak compared to the amplitude of the surrounding artifacts. For example, the amplitude of EOG artifact can reach over 500 μV. The EEG is usually corrupted by artifacts which render its analysis difficult and lead to misinterpretation. For example, in BCI applications, the artifacts can affect the shape of the neural activity and eventually leads to unintentional control of the BCI system [8].

EOG artifact generated by eye blinks or eye movements is an offensive artifact that reduce the quality of the EEG recordings. Its effect is appeared in low frequency (0.5-4 Hz) [9]. EOG generated from eye blinking is a spike with an amplitude that can reach 800 μV and a period of 200-400 ms [10]. However, the EOG originated from the eye movements is characterized by a small amplitude and a large period [11]. The early way consists to identify visually, by an expert, the EEG epoch contaminated by the EOG artifact and then reject them. This approach leads to lose the neural activity and is not automatic. To overcome these inconveniences, the researchers proposed to use artifact handling methods, which aim to remove or reduce the artifact while retaining the neural activity.

Recently, several artifacts handling methods are proposed [12]. Szibbo et al. [13] proposed to remove eye blink artifacts using Savitzky-Golay smoothing filter. They concluded that the developed approach can reach results comparable to those obtained by ICA method (Independent component analysis). Hu et al. [14] developed an automatic and unsupervised technique for detecting and removing of the ocular artifact from EEG recordings. They used spectral entropy and spatial distribution entropy to identify the artifact and canonical component analysis (CCA) to decompose the EEG signal into its correlated components. Thereafter the multichannel Winer filter is used to reduce ocular artifacts in these components. Issa and Juhasz [15] used wavelet transform and ICA to reduce EOG artifacts in EEG recordings. The proposed approach aims to correct the contaminated independent components instead of rejecting them. Maddirala and Shaik [16] applied SSA algorithm on the raw EEG signal with the aim of extracting the EOG reference signal which is needed for the adaptive noise canceler (ANC) to reject the EOG artifact in corrupted EEG signals. Nguyen et al. [17] developed an algorithm for EEG using the time-frequency proprieties of the wavelet transform and neural network. They compared the proposed algorithm with ICA and adaptive wavelet thresholding method. Yang et al. [18]...
used deep learning algorithm to suppress EOG artifacts from EEG signals. They compared their algorithm with SOBI algorithm, ICA, kurtosis-ICA and shallow network algorithm. Le [19] introduced an automated method for removing EOG artifact from EEG signal based on deep wavelet sparse autoencoder. The presented experimental results showed that the developed approach is capable to remove effectively the EOG artifact, while retaining the EEG information. Sun et al. [20] used 1D- residual convolutional neural networks to suppress the EOG artifacts in EEG signal. They showed that the proposed algorithm can reach significant value of signal to noise ratio and preserve the non-linear proprieties of EEG data. Dursun et al. [21] proposed a method based on wavelet transform-based rule and correlation to remove the EOG artifact from sleep EEG recordings. More comprehensive reviews of this topic and extensive bibliographies can be found in [22,23].

In the present work, we propose a fully automatic algorithm to detect and remove the EOG artifact in EEG signals, while preserving the neural activity. This algorithm contains three blocks. The first block aims to detect the EOG artifacts based on three statistical parameters, namely, the entropy, kurtosis, and skewness. These three parameters are used to feed a fuzzy inference system (FIS). Based on these parameters, the FIS takes a decision about the existence or not of the EOG artifacts. Once an EEG epoch is claimed to be artifactual, it will be transferred to the second block. In latter block, the five-level SWT (stationary wavelet transform) is introduced to decompose the EEG epoch into approximation and detail coefficients. To reduce the distortion of the EEG signal, we propose to introduce other fuzzy inference systems in order to select only the contaminated coefficients. Thereafter, the universal thresholding method is then applied to the selected coefficients. This approach allows to preserve intact the uncontaminated coefficients and consequently reduce the effect of the artifact removal. Finally, in the third block, the inverse SWT applies to the uncontaminated and thresholded coefficients to reconstruct the EEG free from EOG artifacts. The proposed technique is evaluated based on the semi-simulated artifactual EEG signal and fully real artifactual EEG signal.

The rest part of this paper will be organized as: the second section presents the proposed algorithm. Sec. 3 illustrates the experimental results. Finally, Sec. 4 shows the conclusion of this paper.

II. PROPOSED ALGORITHM

Figure 1 displays the flowchart of the proposed method. This flowchart can be decomposed on three blocks.

A. Detection block

The detection of the EEG artifacts is done based on the fact that the features of the artifacts are different from those of the EEG signal. The artifact tends to be random while the EEG background is organized. The amplitude of the EEG signal is lower than that of these artifacts. The EEG is symmetric while artifacts are antisymmetric. Based on these facts, it is then possible to develop a system which able to detect the EEG epochs. This can be done in the following steps:

Step 1. Segmentation

In this phase, the incoming EEG signal, $S_n$, is decomposed into $m$ non-overlapped segments, of length $N$. Lets $x_i$ be the $i$th segment.

Step 2. Feature extraction

At this level, each EEG epoch, $x^j$, is represented by a three statistical features, specifically, the composite multiscale entropy (CMSE), kurtosis (K), and skewness (S).

- CMSE is used to measure the randomness of the EEG epoch. It returns lower values for an artifactual epoch and higher for a non-artifactual epoch. For an epoch, $x^j$, CMSE is defined as follow [24]:

$$CMSE(x^j, r, m, r) = \frac{1}{T} \sum_{k=1}^{T} \left( -\ln \frac{n_{k,r}^{m+1}}{n_{k,r}^m} \right)$$

Where $r$ is a scale factor, $m$ is a length of template vectors used to compute the sample entropy of $x_i, r$ is a predefined tolerance, $n_{k,r}^m$ denotes the overall number of $m$-dimensional matched vector pairs, and $n_{k,r}^{m+1}$ is the overall number of $(m + 1)$-dimensional matched vector pairs.

![Flowchart of our approach for artifact detection and removing from the EEG signals.](image)
Step 3. Detection of artifacts

As we can see from figure 1, on the block of detection, these three parameters (CMSE, K, and S) are used as the input of the first decision stage. This latter contains a fuzzy inference system (FIS1). Based on these three parameters, the FIS1 decides if the epoch is corrupted by EOG artifact or not.

The use of FIS1 allows to decide automatically if an epoch is contaminated or not. Therefore, the proposed algorithm is an alternative of several existing algorithms that required the intervention of the user or prior information to identify the artifactual epochs [25-27].

The fuzzy logic theory is used to build the FIS. We use the trapezoidal functions as the membership functions of the three considered variables (CMSE, S and K). They are defined by:

\[
\mu(x, a, b, c, d) = \max \left( \min \left( \frac{x - a}{b - a}, 1, \frac{d - x}{d - c} \right), 0 \right)
\]

Where \(a, b, c, \) and \(d\) are the \(x\) – coordinates of the four corners of the trapezoidal function. We have defined these parameters for each fuzzy logic variable by identifying the variation range of each of them. This is done by calculating the CMSE, S and K for different epochs of simulated artifact-free EEG signal, and simulated artifactual EEG signal. After deep studying of the obtained variation ranges, we have decided to divide the variation range of CMSE into two membership functions, S into three membership functions, and K into two membership functions.

The inputs of our FIS1 are then CMSE, S and K. Its output contains two classes labelled artifact and artifact-free. FIS1 defines then the output class based on the input values (cmse, s and k) and an ensemble of predefined fuzzy rules. In the literature, several types of fuzzy rules have been used. In our study, the general form of fuzzy rules is:

\[
R_j: \text{if } \text{cmse is } A_{j1} \text{ and } s is A_{j2} \text{ and } k is A_{j3} \text{ then class is } C_j \text{ with } \omega_j
\]

Where \(R_j, j = 1,2, ... , L, \) is the \(j\)th rule, \((\text{cmse}, s, k)\) is the feature vector, \(A_{j1}, n = 1,2,3,\) is the fuzzy sets associated with different input variables, \(\omega_j\) is the weight of the \(j\)th fuzzy rule and can be equal either 0 or 1, and \(C_j\) is the output class \{artifact, artifact-free\}.

B. Removal block

Once an epoch is claimed to be artifactual, the next step is to remove artifacts, while preserving the EEG background. As previously discussed in the Sec. I, several approaches have been developed to detect and remove EEG artifacts. In this work, we proposed to use the wavelet thresholding method, which intends to decompose the EEG signal using wavelet transform into approximation and detail coefficients. Thereafter, different coefficients are thresholded to suppress the artifacts. However, in this study, we introduce the wavelet transform to decompose the EEG signal into approximation and detail coefficients, thereafter, we introduced a second fuzzy inference system (FIS2) based detection system in order to select only the contaminated coefficients and preserve intact the artifact-free coefficients (see Fig. 1).

Several types of wavelet transforms (WT) have been proposed such as DWT (discrete WT), CWT (continuous WT), PWT (packet WT), SWT (stationary WT), etc. In this work, we chose to use SWT since it is translational invariant. That means that slow variation in a signal properties cannot create a significant variation in wavelet coefficients and in energy distribution in wavelet scales.

To use SWT, we firstly need to choose the mother wavelet and decomposition level. In this work, we use Haar wavelet as the mother wavelet since it is able to follow the transient artifactual activities better than other mother wavelets. The decomposition level is set based on the bandwidth of EEG signals which is varied from 0.05 to 128 Hz and the useful bandwidth of the considered application. In this study, we consider level-5 decomposition since it allows to cover different EEG frequency bands.

The application of SWT, with Haar mother wavelet and level-5 decomposition, on an EEG epoch \(x^i\), returns the five detail coefficients \(d_{i1}, d_{i2}, ..., d_{i5}\) and final approximation coefficient \(a_{i6}\).

Once the incoming epoch is decomposed using SWT, the next step aims to identify the contaminated SWT coefficients. For that purpose, we extract the statistical features of different coefficients using the three considered statistical parameters (CMSE, S and K). Thereafter, as in the block of artifact detection, we use other fuzzy
inference systems (FIS2) in order to determine the corrupted SWT coefficients. The last step is to denoise the detected contaminated coefficients. In this paper, we denoise these coefficients using non-negative garrote shrinkage function (NNGSF). We choose to use this function due to its interesting properties such as its low sensitivity to the input fluctuations and low bias [28].

NNGSF for a given coefficient $b_{il} \in \{a_{i1}, d_{i1}, d_{i2}, ..., d_{i6}\}$ with $l = 1, 2, ..., 6$ is defined as follows.

$$
g(b_{il}) = \begin{cases} b_{il} & \text{if } |b_{il}| < T_{il} \\ T_{il}^2 & \text{if } |b_{il}| > T_{il} 
\end{cases}
$$

Where $T_{il}$ is the threshold value for the coefficient $b_{il}$ and given as [29]

$$
T_{il} = c_i \sqrt{2 \ln N}
$$

Where $N$ indicates the length of EEG epochs and $c_i$ indicates the estimated noise variance for $b_{il}$ and is defined as [30]

$$
c_i = \frac{\text{median}(|b_{il}|)}{0.6745}
$$

C. Reconstruction block

This block aims to reconstruct the EEG epoch by applying the inverse SWT on the non-contaminated and denoised SWT coefficients. Thereafter, the cleaned EEG signal may be obtained by the concatenation of different denoised and uncorrupted epochs.

III. PERFORMANCE EVALUATION OF THE PROPOSED ALGORITHM

A. Data description

To assist the performance of our algorithm, we considered two types of datasets, semi-simulated and fully real artifactual EEG datasets.

The semi-simulated artifactual EEG dataset is obtained by adding linearly the EOG artifact template to the pure real EEG data. The characteristics of the used real EEG dataset are described in Table 1.

The fully real EEG data are collected from the BCI competition–IV. Table 1 displays the characteristics of used datasets.

B. Metrics for performance evaluation

The performance of the developed algorithm are assisted based on two types of metrics. The first type allows to quantify how much the artifacts have been removed. This type contains: $\Delta SNR$ which measure the enhancement in signal to noise ratio (SNR) and $\lambda(\%)$ which represents the amount of artifact removal in percentage. The second type of metrics is introduced to quantify the distortion brought to the EEG signal due to the artifact removal. This type includes: the enhancement in power spectral density, $\Delta PSD$, enhancement in the root mean square error, $\Delta RMSE$, enhancement in signal to noise and distortion ratio, $\Delta SNDR$, difference in coherency, $\Delta Coh(\%)$, and difference in correlation, $\Delta Cor(\%)$. Correlation function is used to measure the resemblance between two signals in time domain. While coherence function is used to measure the similarity in frequency domain.

C. Results

Figure 2 shows the semi-simulated artifactual EEG signal, the obtained statistical features, and the artifact-free EEG signal as well as the reference EEG signal. The duration of an epoch is 1 s. Figures 2(b-d) show that the three considered statistical features are capable to detect the EOG artifacts. From these figures it is clear that an epoch is artifactual if $CMSE$ is low, skewness and kurtosis are high. These parameters are used in the first detection block, as inputs of FIS1, in order to detect automatically the artifactual epochs. The detected artifactual epochs are decomposed into detail and approximation coefficients using level-5 SWT. Thereafter, the FIS2 introduced in the second block of detection (see figure 1) are used to select the artifactual coefficients. NNHSF is then used to denoise the artifactual coefficients. Finally, the inverse SWT is used to find the artifact-free epoch. Figure 2(e) demonstrates that the proposed approach can successfully detect and remove the EOG artifacts.

The proposed algorithm is also assisted quantitatively using several metrics. Table 2 demonstrates the quantitative performance of our algorithm in the case of fully real EEG data. This figure demonstrates clearly that the suggested algorithm can detect and successfully remove the EOG artifacts.

D. Comparison with other methods

As mentioned in the introduction part, numerous techniques have been developed to detect and remove EOG artifacts from EEG signals [22,23]. In this paper, we choose to compare the performance of our algorithm with wavelet ICA (wICA) [30] which is one of the most used algorithms for EEG artifact removal.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Number of channels</th>
<th>Sampling frequency (Hz)</th>
<th>Bandwidth of a Bandpass filter</th>
<th>Number of subjects</th>
<th>Additional channels</th>
<th>Number of classes</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEG/EOG [31]</td>
<td>19</td>
<td>200</td>
<td>0.5-40</td>
<td>27</td>
<td>EOG</td>
<td>-</td>
</tr>
<tr>
<td>Dataset-1 [32]</td>
<td>64</td>
<td>1000</td>
<td>0.05-200 Hz</td>
<td>7</td>
<td>None</td>
<td>2</td>
</tr>
<tr>
<td>Dataset-2a [33]</td>
<td>22</td>
<td>250</td>
<td>0.5-100</td>
<td>9</td>
<td>3 EOG</td>
<td>4</td>
</tr>
</tbody>
</table>

TABLE 1. Description of the used datasets.
The EEG signal is a very useful tool in medical and technology fields. In medical field, the EEG recordings are used in diagnosis of neurological troubles, prevention of mental diseases. While in technological field, EEG signals are mainly used in BCI. The main issue in EEG applications is that EEG signals are frequently affected by the physiological and non-physiological artifacts. EOG is one of the first origins of artifacts in EEG recordings. In this study, we proposed a fully automatic algorithm. Figures 2 and 3 show that the proposed algorithm can perform well in two considered datasets: semi-simulated and real EEG data. Table 2 shows that our algorithm outperforms the performance of wICA with regard to the amount of artifact removal and distortion brought to the EEG signal. For example, in the case of our algorithm, $\lambda(%)$ (amount of artifact removal in percentage) can achieve 81.57 % and 30.36 % in the case of wICA. The high value of $\lambda(%)$ reached by the proposed algorithm means that this algorithm is able to detect the most EOG artifacts existed in EEG signal. The high value in terms of $\Delta Cor(\%)$ and $\Delta PSD$ demonstrates the capacity of our method to remove EOG artifacts without affecting much the neural activity.

**CONCLUSION**

This paper was concerned with the development of an EOG artifact detector and removal from EEG signals. The presented qualitative and quantitative results demonstrate that our algorithm is performed well in different scenarios. The comparison of the developed algorithm with the wICA method shows the higher efficacy of our algorithms. This algorithm is fully automatic, doesn’t require prior information, doesn’t need of the reference channel, can be used in single and multichannel applications. Based on these advantages, we expected that this algorithm will be useful in different modern applications of the EEG signal, specifically in ambulatory applications of the EEG signal where the artifacts are a serious problem.

![Figure 2. Evaluation of the proposed method in the case of semi-simulated artifactual EEG data.](image)

![Figure 3. Evaluation of the proposed algorithm using fully real corrupted EEG data.](image)

**TABLE 2. Comparison of the proposed algorithm with wICA.**

<table>
<thead>
<tr>
<th>Methods</th>
<th>$\lambda(%)$</th>
<th>$\Delta SNR$</th>
<th>$\Delta RMSE$</th>
<th>$\Delta PSD$</th>
<th>$\Delta SNDR$</th>
<th>$\Delta Cor(%)$</th>
<th>$\Delta Coh(%)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>wICA</td>
<td>30.3602</td>
<td>1.822</td>
<td>17.684</td>
<td>-263.251</td>
<td>-3.694</td>
<td>117.305</td>
<td>1.022</td>
</tr>
<tr>
<td>Proposed</td>
<td>81.57</td>
<td>7.6187</td>
<td>57.8196</td>
<td>97.0964</td>
<td>-4.6396</td>
<td>402.0621</td>
<td>1.7037</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENT

This work was funded by Ministry of High Education, Scientific Research and Innovation, The Digital Development Agency (DDA) and the CNRST of Morocco under the number 14/FSA/2021.

REFERENCES

Real-time Gait Trajectory Prediction Based on Convolutional Neural Network with Multi-head Attention

1st Jie Yin
Department of Automation
Tsinghua University
Beijing, P.R.China
yinj18@mails.tsinghua.edu.cn

2nd Tao Xue
Department of Automation
Tsinghua University
Beijing, P.R.China
xuetao940810@163.com

3rd Tao Zhang
Department of Automation
Tsinghua University
Beijing, P.R.China
taozhang@mail.tsinghua.edu.cn

Abstract—The lower limb exoskeleton can effectively improve the ability of users. Accurate gait trajectory prediction can enhance the effect of lower limb exoskeletons. The current gait trajectory prediction methods have the disadvantages of insufficient prediction accuracy and long calculation time. This paper proposes a convolutional neural network model with multi-head attention to predict the gait trajectory. Compared with the widely used recurrent neural network model, the model we proposed in this paper can predict gait trajectory with higher accuracy and shorter calculation time. The error is reduced by up to 21.3% and the average calculation time is reduced by 62.8%. Based on the proposed model, we design a controller of the lower limb exoskeleton to achieve better effects in mixed gait and eliminate the delay.

Index Terms—exoskeleton, gait trajectory prediction, deep learning, attention

I. INTRODUCTION

The lower limb exoskeletons that assist in improving athletic ability play a crucial role in medical rehabilitation, assistance for the elderly and disabled, and industrial production [1][2][3][4][5]. People use lower limb exoskeletons to complete the more delicate and complex tasks, the more precise controllers people need. Otherwise, the lower exoskeleton cannot fully utilize its hardware capabilities and may even cause danger.

Some lower limb exoskeleton control strategies are based on human dynamics. In [6] Ren et al. use the inverse dynamics model of the body to predict segmental motions and ground reactions. In [7] Aguirre-Ollinger et al. propose an active-impedance control method based on the human body impedance model. Zhang et al. [8] present a control method of lower limb exoskeleton based on admittance shaping. However, the structure of the human body is exceedingly complex, and the differences between individuals are enormous. These model-based methods are impracticable to be universal among different individuals. These methods rely on an accurate dynamic model and complete state measurement. The lower limb is a rigid-flexible coupled dynamic model, and the model approximation will cause extra errors.

Currently, the most widely used lower limb exoskeleton control method is the gait phase estimation method. Its basic idea is to get the current phase of the gait and then control the lower limb exoskeleton relying on the mapping function between the gait phase and the auxiliary force, which is normally a sine function. Pappas et al. [9] and D.Kotiadis et al. [10] directly measure the gait phase by processing the signals of many specialized sensors. Kang et al. [11] use a deep neural network model to calculate the gait phase, reducing equipment dependence. The gait phase estimation method relies on the mapping relationship between the gait phase and the auxiliary force, which often cannot be customized to meet the different ancillary requirements of varying people. In addition, there is often a delay in the gait phase correction process. During the delay period, the controller outputs the auxiliary force according to the incorrect phase, which is likely to cause danger. The gait phase estimation method is suitable for periodic gait assistance, but it has insufficient adaptability to mixed gait.

Lim et al. [12] propose delayed output feedback control to control the robotic exoskeleton. This control method does not rely on gait phase or environment recognition, but it is not suitable for mixed gait.

Another type of lower limb exoskeleton control method is the gait trajectory prediction method, which relies on the past gait trajectory to predict the future gait trajectory to control the auxiliary force. Glackin et al. [13] predict the user’s trajectory based on Gaussian process analysis. Liu et al. [14] and Su et al. [15] use the LSTM deep neural network model to predict the trajectory and achieve better results. Trajectory prediction is a control method that can handle non-periodic gait with little delay. The main limitation of its control effect is the prediction accuracy.

The LSTM neural network model has disadvantages of...
accumulated errors and long cycle calculation time. The prediction error of gait trajectory limits the hardware capability of the lower limb exoskeleton. Delay caused by calculation weakens the effect of the lower limb exoskeleton and even causes danger. These two shortcomings become more and more serious with the increase of the length of the prediction sequence. Transformer-based deep architecture has shown great potentiality in time series prediction [16].

In [17] Martinez-Hernandez et al. present a novel learning architecture for the recognition and prediction of walking activity and gait period, which is composed of a Convolutional Neural Network (CNN), a Predicted Information Gain (PIG) module and an adaptive combination of information sources.

This paper proposes a convolutional neural network model with multi-head attention [18], which uses the human operator’s past thigh flexion angle sequence as the only input to predict the future thigh flexion angle sequence. The proposed model improves the data utilization to improve the prediction accuracy and avoids autoregressive calculation to reduce the calculation time.

Based on the proposed model, this paper designs a type of lower extremity exoskeleton controller, which can achieve better effects in mixed gait and eliminate the delay. Experiments prove that our controller can achieve better results, compared with the delayed output feedback control method.

II. MULTI-HEAD-ATTENTION CNN MODEL

The multi-head-attention CNN model for gait trajectory prediction is shown in the Fig. 1. The model is divided into three parts: the input network, the CNN loop network, and the output network. This model uses the past N-length thigh flexion angle sequence $X = [t_1, t_2, t_3, \ldots, t_N]$ to predict the future P-length thigh flexion angle sequence $Y = [t_{N+1}, t_{N+2}, \ldots, t_{N+P}]$.

The input network is a $4 \times L$-output-channel convolutional network with a gated liner unit (GLU) activation function. It expands the dimension of $X$ which is in the form of $(1, N)$, into multi-dimensional data $Y_0$ which is in the form of $(2 \times L, N)$, to extract more features from single-dimensional time-series information $X$.

The CNN loop network includes two parts. The first part is a $2 \times L$-output-channel convolutional network with a GLU activation function, and its output named $Y_1$ is in the format of $(L, N)$.

The second part of the CNN loop network is the self-multi-head attention network. The core of the multi-head attention mechanism is the scaled dot product attention. Calculating the scaled dot product attention requires three parameters: queries($Q$), keys($K$), whose dimensions are $d_k$, and values($V$), whose dimension is $d_v$. To self-attention, the three parameters are the same input data. The formula is as follows:

$$Attention(Q, K, V) = softmax\left(\frac{Q \cdot K^\top}{\sqrt{d_k}}\right) \cdot V \quad (1)$$

Compared with single attention, multi-head-attention that independently calculates parts of the attention and merges it into the whole can get better results. Suppose the number of heads of the multi-head attention network is $H$. The formulas are as follows:

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \quad (2)$$

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_H) \cdot W^o \quad (3)$$

Among (2) and (3), $W^Q$, $W^K_i$, $W^K_i$, $W^V_i$ are trainable parameter matrices.

The self-multi-head-attention network takes $Y_1$ as input and gets $Y_2$. $Y_2$ has the same form as $Y_1$. $Y_2$ and $Y_1$ are merged to obtain new data named $Y_3$ which is in the form of $(2 \times L, N)$. $Y_3$ is used as a new input to perform a new loop and this loop performs for $M$ times. The self-multi-head-attention network merge its input and output to achieve higher accuracy and use less computing space. The result of the loop passes through a $P$-output-channel CNN network with to get $Y_4$ which is in the form of $(P, N)$.

The output network is a multi-layer perceptron at the end of the model. $Y_4$ is passed through the output network to obtain the predicted thigh flexion angle sequence in the
future named \([T^*_N+1, T^*_N+2, \ldots, T^*_N+P]\), and calculate the loss function \(\text{loss}(T^*_N+1, T^*_N+2, \ldots, T^*_N+P)\) with real data \(Y\). The loss function is composed of Mean Absolute Error (MAE) (4) and Mean Absolute Derivative Error (MADE) (5).

\[
\text{MAE}[T^*_N+1, T^*_N+2, \ldots, T^*_N+P] = \sum_{x=N+1}^{X=N+P} |T^*_x - T_x| \quad (4)
\]

\[
\text{MADE}[T^*_N+1, T^*_N+2, \ldots, T^*_N+P] = \sum_{x=N+2}^{X=N+P} \left| (T^*_x - T^*_{x-1}) - (T_x - T_{x-1}) \right| \quad (5)
\]

\[
\text{loss}(T^*_N+1, T^*_N+2, \ldots, T^*_N+P) = \text{MAE} + \text{MADE} \quad (6)
\]

III. CONTROLLER

Based on the convolutional network model with multi-head attention proposed by this paper, we propose a controller of the exoskeleton. In our controller, the auxiliary force of the lower limb exoskeleton, named \(F\), is dependent on the difference between the predicted thigh flexion angles. The magnitude of the force is proportional to the absolute value of the difference. The sign of the difference determines the direction of the auxiliary force. We pass the difference through a low-pass filter to get smooth force.

\[
F(x) = \text{filter}[\alpha \ast (T^*_{x+1} - T^*_x)] \quad (7)
\]

In (7), \(\alpha\) is the gain factor. The order of the low-pass filter selected in this paper is 1, and the normalized cut-off frequency is 0.05 Hz.

Since our model can predict walking and stationary gait trajectories, our controller can control in both walking and stationary states.

Both calculation and auxiliary force generation take time, usually in milliseconds. Our data collection frequency is 100Hz, which means that applying the predicted angle values at time \(N\) and \(N+1\) to control the lower extremity exoskeleton inevitably causes delay. To solve this problem, we predict the thigh flexion angle value \(T^*_N\) and \(T^*_N+2\) at time \(N\) to calculate \(F(N+1)\) early. By doing so, our controller eliminates the delay.

We multiply the difference between \(T^*_{x+1}\) and \(T^*_x\) and the difference between \(T_N\) and \(T_{N+1}\) to represent the momentum value generated by the lower extremity exoskeleton as (8). Positive values represent the force and the movement are in the same direction, and negative values represent the force and the movement are in the opposite direction.

\[
\text{MOMENTUM}(x) = (T^*_{x+1} - T^*_x) \ast (T_{x+1} - T_x) \quad (8)
\]

IV. EXPERIMENT AND RESULT

In our experiment, \(N = 80\), \(P = 10\), \(L = 30\), \(H = 4\), \(\alpha = 1\), which means that our model uses the past 80-length thigh flexion angle sequence to predict the future 10-length thigh flexion angle sequence with 4-head attention. And the core length of the convolutional network is set to 5, and \(M\), the time number of main loops, is 2. The total number of trainable parameters of the network is 98301. In the training stage, we use the Adam optimizer, and a fixed learning rate of \(1 \times 10^{-4}\) is set to ensure the best effect.

A. Lower-limb Exoskeleton

Our team has a lower limb exoskeleton system Ares V3.0, which weighs 3kg (see Fig.2). It is composed of a waist control package, hip joint drive joint, and leg flexible link. The waist control package has a built-in servo driver and IMU sensor. The flexion/extension is the exoskeleton’s active degree of freedom, providing auxiliary force in the sagittal plane. The controller obtains information such as joint angles in real-time, and the control frequency is 1kHz.

B. Data Collection

We fix the sensor system on the outer thigh of the tester with a strap. The sensor system integrates an inertial measurement unit to measure three-axis acceleration and three-axis angular velocity. After complementary filtering, the participant’s leg posture can be obtained, and the data can be transferred to the computer.
The locomotion mode recognition experiments were conducted with 7 healthy subjects (one female, six males; 25.2 ± 2 years old; 172.0 ± 5 cm height; 70.10 ± 20 Kg weight).

All the participants have given their consent before taking part in the study. The experimental protocol was approved by the Institute Review Board of Tsinghua University (No. 20200014).

We collected 7 sets of data from 7 participants. We named each participants’s data as D1, D2, D3, D4, D5, D6 and D7. D1, D2, D3, D4, D5, and D6 are the thigh flexion angle sequences recorded by 6 participants performing slow walking, fast walking, going upstairs, going downstairs, sitting down, and standing up. D7 is the thigh flexion angle sequence of the seventh experimenter walking at variable speeds. We collect D7 to test the generalization ability of our model.

All data were min-max scale to be between 0 and 1 to get better results.

### TABLE I

<table>
<thead>
<tr>
<th>Cross Validation Results</th>
<th>mean absolute error (degree)</th>
<th>mean absolute derivative error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>D2</td>
<td>D3</td>
</tr>
<tr>
<td>LSTM model</td>
<td>2.0336</td>
<td>2.0158</td>
</tr>
<tr>
<td>traditional CNN</td>
<td>1.8350</td>
<td>1.7810</td>
</tr>
<tr>
<td>CNN model</td>
<td>1.7901</td>
<td>1.5856</td>
</tr>
</tbody>
</table>

Fig. 4. Prediction of slow walking data in D3

Fig. 5. Prediction of fast walking data in D3

Fig. 6. Prediction of going upstairs data in D3

Fig. 7. Prediction of going downstairs data in D3

Fig. 8. Prediction of sitting down data in D3

Fig. 9. Prediction of standing up data in D3
C. Baseline

For comparison, this paper builds one LSTM network model with attention to complete the same prediction task (see Fig.3). This model adds an attention network to the traditional LSTM model. Firstly, the attention weight is calculated with the input and hidden state of the decoder, and then the information is extracted from all hidden states of the encoder according to the attention, finally we input the information into the decoder to calculate the new hidden state and output. The loss function is the same as the model proposed in this article. We set the parameters of the LSTM network reasonably to achieve the best results. The number of trainable parameters is 87301, which is in the same order of magnitude as the number of parameters of the model proposed in this paper. In the training stage, we use the Adam optimizer to the LSTM, and the learning rate is set to be $1 \times 10^{-4}$ to ensure the best effect.

We also apply the delayed output feedback control (DOFC) method to compare with the controller of the exoskeleton proposed in this paper. In the DOFC, auxiliary force $F(x)$ is calculated according to (9), in which $d$ indicates the length of the delay.

$$F(x) = \alpha \times \sin(T_{x-d})$$  \hspace{1cm} (9)

D. Result

Data D1 is used as valid data, D2, D3, D4, D5, and D6 are used as test data in turn, and the rest are used as train data. If the model cannot get better performance after at least 3 episodes, we stop training and record the parameters that perform best on valid data. The results are shown in the table.

Compared with the LSTM network with attention, the convolutional neural network with multi-head attention proposed in this paper has a significant improvement in prediction accuracy. It is because the cumulative error is eliminated in our model. In the test, the convolutional neural network with multi-head attention proposed in this paper takes an average of 32.4s, and the LSTM with attention takes an average of 86s. It can be seen that because of the parallel calculation of the convolutional neural network, the model proposed in this paper is faster and meets the real-time requirements better.

The results are shown in the Fig.4, Fig.5, Fig.6, Fig.7, Fig.8 and Fig.9. In order to show the results more comprehensively, several prediction results are spliced together to form a complete prediction curve. It can be seen that our model predicts...
more accurately when the gait is stable and adjusts faster when the gait changes rapidly.

In order to prove the generalization performance of the convolutional neural network with multi-head attention proposed in this paper, we take D7 as test data. Predicted by the model trained by D1, D2, D3, D4, D5, and D6, the absolute mean error is 1.8020, and the absolute mean derivative error is 0.5234.

The result is shown in Fig.10. It can be seen that the model predicts the unknown and dynamically changing gait very well, which proves that the model has good generalization ability.

We change the parameter $P$ to 2, which means the model predicts the future 2-length thigh flexion angle sequence, and use D1, D2, D3, D4, D5, and D6 as the train data to train the model. We apply the controller proposed in this paper and DOFC on the D7 data.

The obtained auxiliary force is shown in Fig.11. It can be seen that the force produced by our controller keeps pace with the gait, even when the gait changes rapidly, which is better than the force produced by DOFC.

Momentum histograms are shown in Fig.12. Under the controller proposed in this paper, the ratio of negative momentum to positive momentum is 5.5%. Under the DOFC, the ratio of total negative momentum to total positive momentum is 58.8%. It can be seen that the lower limb exoskeleton using our controller can effectively assist, and there are very few cases of hindering movement. The control strategy based on the stable gait such as DOFC is hard to adapt to the dynamic gait.

White noise is added to D7 to make the signal-to-noise ratio (SNR) be 10, 5 and 1. The experimental results are shown in the table II. It can be seen that our proposed controller achieves good results even in the presence of noise.

V. CONCLUSIONS

This paper proposes a convolutional neural network model with multi-head attention for gait trajectory prediction. This model uses a multi-head attention mechanism to break up the spatial barriers between convolutional neural network data and effectively improve data utilization.

Compared with the widely used LSTM network in time series information processing, our model can predict future time series information better and faster. The model proposed in this paper has considerable generalization ability, and can also predict the gait trajectory well for the types of actions that have not been encountered in the train data. The improvement of the model in this paper will greatly promote the control method of lower limb exoskeleton relying on trajectory prediction.

This paper also proposes a lower limb exoskeleton controller based on our trajectory prediction model. Based on accurate trajectory prediction, experiments have verified that this control method can effectively assist and effectively reduce resistance even in the presence of noise. And this control method eliminates the delay by calculating auxiliary force $F(x)$ early. Lower limb exoskeleton control strategies based on gait trajectory prediction have great potential for future development.

REFERENCES

Designing and manufacturing an Android-controlled robotic arm using rapid prototyping

Alex Marnell, Mahmood Shafiee, Amir Hosein Sakhaei
Mechanical Engineering Group, School of Engineering
University of Kent
Canterbury CT2 7NT, United Kingdom
apm42@kent.ac.uk; m.shafiee@kent.ac.uk

Abstract—This paper aims to design and manufacture an Android-controlled robotic arm for the purpose of picking and placing objects in constrained environments. The motivation of the research is to assess the feasibility of controlling complex real-time operating systems using a personal handheld device, with a future scope to control larger scale operations in the same way. Our robotic arm is produced using rapid prototyping techniques and controlled by an Arduino based development board. It receives control commands from a bespoke Bluetooth app, allowing to select different servo operating modes. The system will be fully controllable from the Android device, allowing the user to manually adjust the position of the arm or running predefined algorithms in automatic mode. Experimental tests are conducted to evaluate the practical limitations of the robotic arm design coupled with the PCB circuitry used as a shield to control the system. The test results show that the system is capable of lifting weights up to 80g with a current draw of 970mA. The automatic control algorithm is found to operate for up to 12 minutes before encountering system bugs or overheating and proves to be a highly repeatable test system with a high level of adaptability. However, the 3D printed PETG is seen to wear over time; and therefore, a more suitable material is recommended to be utilized for fabrication.

Keywords—Robotic arm; CAD/CAM; 3D printing; rapid prototyping; Android controller.

1. INTRODUCTION

The robotic industry is one of the fastest growing global markets with applications ranging from automobile, to aerospace and biomedical. Robotics is a multidisciplinary area that involves the design, manufacture, control, and programming of robots [1]. In recent years, the focus of the robotic industry has been on improving workplace safety by reducing human involvement in dangerous environments such as x-ray radiology rooms, deep-sea installations, underground mines, nuclear power stations, etc.

Within robotics, the field of robotic arms has received increased attention due to the trend for building more advanced and automated mechatronic systems. The robotic arms are widely used in industry, for example for painting car bodies, die-casting, spot welding, packaging and assembling products. The precision offered by robotic arms, in combination with their ability to work around the clock, have allowed companies to improve their quality standards, reduce safety risks associated with manipulation of heavy components, and cut down on the amount of waste material produced in the manufacturing process.

The complexity of a robotic arm is usually proportional to what is referred to as Degree of Freedom (DoF). DoF represents the direction in which a robot can make movements. Higher DoF offers more diverse motion and therefore more control over the operation being performed, with the tradeoff being an increased power draw and more complicated control systems. The most common style of robotic arms is designed to emulate the human arm, allowing three DoF in the form of a shoulder, elbow, and wrist joint. This results in full manoeuvrability and manipulation of their process within the operational radius of the system. Different control systems have been developed alongside the development of robotic systems. These control systems vary in complexity to match the needs and applications of different robotic arms. While most of these systems have purpose-built control units, it is more user-friendly to design small handheld devices such as a smartphone to control such robots in real-time. The smartphones are nowadays as computationally powerful as the computer systems used by NASA during the Apollo era of space flight and exploration. They can control robotic systems wirelessly via a Bluetooth network.

A lot of studies have been conducted to model the dynamic behavior of robotic arms. Many different tools have also been developed to send user commands to a Bluetooth module and control the robot wireless. One of the most common tools to use is the App Inventor 2, developed by MIT (https://appinventor.mit.edu/). This coding environment is popular due to its accessibility and ease of use. It provides a block-based coding suite where functions are connected to operators on a graphic interface. This allows for the quick implementation or adjustment of the system code as reported by Ahamd et al. [2]. Szolga et al. [3] developed a bespoke Android app for the control of a three joint robotic arm printed out in polylethylene terephthalate glycol (PETG) material. The communication between the robotic arm and the app was established through Bluetooth protocol. A vacuum pump was mounted at the tip of the robotic arm to handle organic solar cells.

Many different approaches have been reported in the literature on how to produce the torque required for the robot’s movement. While all authors chose to use dedicated motors, the number as well as the types of motors included within the design vary. For example, Rajpar et al. [4] used four motors, three of which were used for positioning and orientation and the fourth one was used to carry out the pick-and-place task. This proved beneficial as the designed system could accurately move to any specified location within the 3D matrix of its reach while monitoring its own position with internal circuitry. Baby et al. [5] used DC
motors to produce torque and drive an end-effector; however, the DC motors required an external monitoring circuit to monitor their position as well as an external motor driver circuit to allow operation in two directions. Qi et al. [6] developed mathematical models to assist the design and 3D printing of a robotic arm. Even though they did not provide many details about the Bluetooth communication process, they presented a methodology for designing a robotic arm system for the purpose of pick and place operations. The paper included relevant test data to confirm the operation of the system and the limits of its operation. Chavan et al. [7] presented mathematical modelling of the 3D trajectory of a robotic arm system. Although the process of designing some vital components such as the motors and body was discussed in detail, there was little information about the implementation of the system and the test results.

The aim of this paper is to design and manufacture a Bluetooth controlled robotic arm for the purpose of conducting pick and place operations. The motivation of the research is to assess the feasibility of controlling complex real-time operating systems using a personal handheld device, with future scope to control larger scale operations in the same way. Our robotic arm is produced using rapid prototyping techniques and controlled by an Arduino based development board. The system sends control data from a bespoke control app on an Android device and receives control commands by a Bluetooth control module interfaced with a microcontroller. It then uses motors to support the orientation and gripping functions of the robot. Experimental tests are conducted to evaluate the practical limitations of the robotic arm design coupled with the PCB circuitry used as a Shield to control the system. The test results show that the system is capable of lifting weights up to 80g with a current draw of 970mA.

The organization of the remainder of the paper is as follows. Section 2 presents the robot’s hardware and software components; Sections 3 reports the test results and discusses the findings; and finally, Section 4 concludes the work.

II. ROBOTIC ARM DEVELOPMENT

A. System’s block diagram

The block diagram of the robotic arm is shown in Figure 1. The system can be seen as three separate units, including the Android based Bluetooth controller app, the Arduino based PCB, and the rapid prototyped mechanical arm that houses the servo motors. The Android device is connected to the Arduino board via a wireless Bluetooth connection with the HC-06 Bluetooth module operating at 2.4GHz. It receives data sent to the system based on user inputs on the app that represent either a new angle selection, mode selection, or reset functions. This data is processed by the ATMEGA328P MCU after receiving it through its Serial Port, and then uses Pulse Width Modulation (PWM) to control the motion of the servo motors within the main structure of the arm. These servos act as pivots within joints, allowing for motion to occur linearly outwards from a central focal point that can be rotated around as a central axis. This in combination with a servo motor located on the end effector to drive the gripper, allowing to execute pick and place operations.

Figure 1. System’s block diagram.

B. CAD model

The CAD model of the robotic arm was developed by the Autodesk Fusion 360 software, and it is shown in Figure 2. The design provides a solid base to support the weight of the system in motion. It includes an open housing for the PCB which is stored underneath the top plate. The servo motor that allows rotation of the system is mounted through the top plate and secured by a screw on the underside, all of which is then securely fastened by the holding plate so that no torque is lost due to slippage of the material. This servo, and each after it, are encased in a purpose-built housing designed to press-fit the servo motors inside, which allow the connection of further components while providing a suitable structure to support forces produced during motion.

Figure 2. CAD model of the robotic arm.

C. Breadboard design

A circuit is set up to run the Arduino Sweep code and ensure that the servo motors are working properly. The circuit is originally designed to power the servos directly from the board which caused frequent errors in the motors due to the lack of sufficient power supply; however, this was quickly rectified with the addition of a separate 9V battery. The battery output was regulated with a LM7805 5V regulator and appropriate smoothing capacitors to allow the servos to be supplied with the current they needed. The HC-06 module was then connected to the
circuit, with the RX and TX pins of the module being connected to the TX and RX pins (serial input pins) of the Arduino board respectively. This allowed for some basic testing commands to be sent to the MCU to operate the servos remotely. A push button was then added to the circuit to allow for a manual interrupt and reset of the system in case if any error occurs. Following this, a switch was added to the 9V power supply to increase the safety of the circuit by allowing the power supply to be manually turned on and off by the user. Finally, two LED indicators were added with some foresight to allow the user to see from a distance whether the circuit is powered, and if the system was in a reset state. The partially complete circuit is shown in Figure 3.

The breadboard allowed for a quick prototyping of the circuitry without having to commit to a single design; however, issues arose with the operation of the servos and smaller components. Due to the number of connections between various components and the board, it became hard to track connections between the jumper cables, slowing down the design process if troubleshooting is needed to occur. These cables would also frequently come loose during transport and storage, or even installation of new components, and would cause errors within the circuit due to short circuits. However, once the circuit is assembled and tested for basic functionality the PCB can be designed.

D. Eagle PCB implementation

Following the circuit prototyping on the breadboard, the PCB schematic is designed using Autodesk Eagle to allow the fixed connectivity between the circuitry and MCU. The first PCB replicated the design of the breadboard circuit in the form of a Shield that could be connected to the MCU via the header pins of the development board. The male header pins would be underneath the Shield and therefore needed to be soldered on the top layer, while the main circuit would be installed on the top of the board and soldered on the bottom layer. The board was designed so that the HC-0 module and servo motor connector pins would be easily accessible with suitable clearance to allow easy installation. This extra clearance between vital components also helped to reduce likelihood of signal interference occurring within the circuit. Lastly, a ground plane was added to the bottom layer of the board, allowing for a common ground between the 9V battery, the board and all the connected circuitries. The final shield board layout is shown in Figure 4.

After fabrication, the pin holes and vias were drilled with a high-speed PCB drill machine and all the components were soldered into place. Then, a second PCB was designed to allow the removal of redundant parts from the development board, as well as allowing all the components to be powered by the 12V input that supplied the board instead of requiring a separate power supply to supply current for the motors (see Figure 5).

E. Rapid prototyping and assembly

The robotic arm structure was manufactured using rapid prototyping techniques. The individual pieces were either 3D printed in PETG or laser cut from acrylic. The 3D printing allowed for more complex designs as the parts could be manufactured as a single piece, as opposed to having to print and assemble multiple smaller pieces out of metal or laser cut parts. The developed system for the Android controlled robotic arm is displayed in Figure 6.
F. Software implementation

The testing of the servos was performed by the Sweep demo code provided with the Arduino IDE. This code initializes a servo after including the Servo.h library and runs it through every position between 0 degrees and 180 degrees and back again. A Bluetooth control software was created to allow a simple implementation of the system. It receives a single letter character from the Bluetooth app and runs conditional statements to drive the servos based on this input. For Bluetooth data to be received by the MCU, a serial port is set up to receive data from the HC-06 module. Once the board is powered on, the MCU runs the setup code for the program and initializes the serial port with Serial.begin(9600), setting the baud rate for the input to match the baud rate of the Bluetooth module output. Within the main loop of the code for the manual operation, the system is set up to run the conditional control statements whenever there is data available on the serial port using: while (Serial.available()>0). This allows for better power management of the system as the main conditional control statements only run when a command is received, otherwise it sits in an idle state. When a command is received, the input is stored as a character using inputByte = Serial.read() and checked against a series of conditional “if” and “else if” statements to tell the motors where to move.

During the initial implementation of the board and its Bluetooth functionality, a prebuilt Android app that could act as a Bluetooth controller was used. This app allowed connection between the Android device and the HC-06 module and was able to send single characters to the MCU by pressing a button on the GUI. Once the initial tests were complete, App Inventor 2 was used to design the GUI and algorithms of the app, allowing direct control of the system from the Android device. Figure 7 shows the GUI built for the robot arm.

Once the “Connect Bluetooth” button is pressed, the GUI displays the available devices and if a device is selected it will start to set up a connection based on the stored name and address. The button text will then be updated to say “Connected” and an internal clock will be initialized. This clock is used as a buffer for the Bluetooth communication, giving the MCU adequate time to carry out the operations without suffering from packet loss due to too much data being transmitted. Once the clock is set, the algorithm will check the state of the global flag and update the system accordingly. Hence, the clock allows for the system to run smoothly and make only one operation at a time. On the rising edge of the clock, if the flag is set to true it will update the display on GUI to match the chosen position for the selected servo. The value is then sent via Bluetooth and the flag is set to false again, thus setting the system up to receive a new update. At any point, the servo operations can be selected by pressing any of the relevant buttons on the GUI. If this is done, then a predetermined integer value is sent to the MCU that corresponds with a servo selection algorithm on board. Once the design of the app was completed and the basic control algorithm produced the desired outputs, it became clear that calibration of the servo motors would be important for the correct setup of the fully integrated system. A small Arduino program was written to allow for the positions of the servos to be controlled by passing an integer value from the Serial Monitor of the Arduino IDE to the MCU. Each servo was then individually connected to the board and manually set to a position of 90 degrees to act as a midpoint for operation (with the limits being 0 and 180 degrees). With the system calibrated and the app fully operational, the main Arduino code could be updated to reflect the new system design.

III. RESULTS AND DISCUSSION

As there are many systems involved in the operation of the robotic arm, some tests are conducted to assess the practical performance of the system. These tests include a lifting capacity test, a connection range test, a current draw test, and a system reliability test. These are briefly explained below.

A. Lifting capacity

The lifting capacity of the robotic arm was tested by gripping some calibrated weights ranging from 10g to 200g. An example of such test is shown in Figure 8, with a 20g weight being lifted. These tests are carried out in a non-destructive manner, and they stop when the servo motors begin to stall.

Figure 7. Final GUI.

Figure 8. The robotic arm lifting a 20g weight.
The results show a good linear relationship between the increase in test weight and the current draw required to produce motion, as can be seen in Figure 9. The system was recorded to have an idle current draw of 80mA when supporting its own weight. This is due to the servos constantly monitoring their own position and drawing the required current to maintain their angle. The current draw then increases up to a maximum current draw of 960mA when lifting an 80g weight.

![Figure 9. The current draw of the robotic arm when lifting weights.](image)

**B. Connection range**

The HC-06 module is rated to allow Bluetooth communications for up to a distance of 10m; however, the performance of this module was tested in a laboratory environment. To test whether the system was connected, two visual indicators were used, including a yellow reset LED that is activated in response to a button being pressed on the GUI, and a red surface mount LED on the Bluetooth module that will begin to blink if the connection is lost. During the test, the connection was tested at one-meter intervals along the length of the laboratory starting at one meter away and found that the system remained connected beyond its specified technical limits with a range up to 62 meters, exceeding initial expectations.

**C. Idle current draw**

The idle current draw was tested to assess the efficiency of the system while idling (not in motion) at intervals within each servo’s range of motion. To do such test, a multimeter was connected serially with the power supply output and the input of the Shield board to measure the current drawn when the system is in a series of idle positions. The test consisted of setting the system to the default state and selecting a single motor to operate. This motor would be set to its upper limit (whether it be digital limits of the motors, or mechanical limits due to design constraints), rounded to the nearest 10 degrees to increase reproducibility and uniform measurements. Commands would be sent from the GUI to the system to increase the angle of the servo by 10 degrees and the current draw displayed on the multimeter would be recorded. This was repeated for the three motors responsible for the motion of the system, as the gripper motor is not influenced by the torques generated in different orientations due to maintaining its alignment with the use of the rocker and struts.

The results show that the Servo 1 has the largest range of motion due to its lack of physical restrictions, whereas Servos 2 and 3 have decreasing range of motion. By performing standard deviation analysis, the test data was used to assess the uniformity of the current draw for each servo motor. The standard deviations for Servos 1, 2 and 3 were calculated as 2.27, 22.48 and 38.26 respectively. This data shows that the Servo 1 has the most uniformity with its power consumption, with the average variation across its operational range being 2.27mA. By comparison, Servo 3 produced the most variation in its power draw, with the average variation in values being 38.26mA within its much more restricted range. This suggests that Servo 1 has the highest overall power efficiency.

An automated system is often required to operate for long periods of time without failing to achieve desirable results. To assess the reliability of the system, the robot arm was run in automatic mode until either of software or hardware components fails. The test was performed five times at two different input voltages, with the system being fully reset using the designated push button of the development board. To allow the system to run unimpeded, it was powered by a variable workbench power supply with a current limiter set to above 1.5A so that the robotic arm could fully regulate its own power. The results show that the system operates best with an input voltage of 9V, being able to operate fully autonomously for over 12 minutes before errors occurred. However, the system was found to begin overheating after a longer period of operation, with the 5V regulator dissipating the 4V excess from the input as heat to its surroundings. The voltage output of the regulator was analyzed by an oscilloscope as shown in...
Figure 11. It caused a lot of noise to be passed into the system when the servo motors required a sudden increase in current draw. However, it is believed that the system eventually failed in test 1 and 2 due to a software bug as the automatic algorithm began to fail with only parts of the movement being complete in each loop.

![Figure 11. Oscilloscope display of voltage output.](image)

Test 3 also failed early due to mechanical wear on the PETG material. A support limb became detached and was repaired with two-part epoxy to create a secure bond to allow for further testing. While undesirable, it highlighted aspects of the design that would need to be improved to increase the longevity of the system. Tests 4 and 5 saw the motors beginning to show signs of stalling as the algorithm continued to run. Due to the prolonged use during testing, the 5V regulator became increasingly hot and as such the efficiency of its power transfer began to become limited as the impedance of the system grew. It eventually restricted the current flow to the servos, causing the eventual failure of the system.

A 7V input was used for tests 6 through 10 to assess the feasibility of using a lower voltage input to reduce the impact of overheating on the functionality of the system. Table 1 shows a relative uniformity and repeatability of the system with the range of the times being 59 seconds, however the effectiveness of the automated algorithm reduced when operating at the lower voltage with the longest operational time being 4 minutes shorter than that of the 9V operation. The main errors identified with the system failure at this voltage input were those relating to the servo stalling or being underpowered and the algorithm ultimately not running effectively.

![Table 1. Automatic run time at different voltages](image)

<table>
<thead>
<tr>
<th>Test number</th>
<th>Voltage input (V)</th>
<th>Run time (minutes)</th>
<th>Cause of failure</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>00:12:18</td>
<td>Software bug</td>
</tr>
<tr>
<td>2</td>
<td>9</td>
<td>00:11:39</td>
<td>Software bug/Overheating</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>00:07:53</td>
<td>Mechanical failure of a limb</td>
</tr>
<tr>
<td>4</td>
<td>9</td>
<td>00:11:53</td>
<td>Stalling motors</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>00:10:11</td>
<td>Stalling motors</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>00:03:57</td>
<td>Software bug</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>00:04:18</td>
<td>Software bug</td>
</tr>
<tr>
<td>8</td>
<td>7</td>
<td>00:03:42</td>
<td>Software bug</td>
</tr>
</tbody>
</table>

In summary, the system operates most efficiently and effectively when powered with the 9V supply and only begins to fail due to software issues that could likely be resolved by adapting the control algorithms. The repeatability of the system proves quite uniform at both power levels with the servo motors failing before any other parts of the robotic system.

IV. CONCLUSION

In this paper, an Android controlled robotic arm was designed and implemented with the ability to be user-programmable and run by an automatic algorithm. A Uno R3 was used as the development board, and a custom Shield PCB contained the operational circuitry. Four MG946R servo motors were used to produce accurate movement of the robotic arm. The robotic arm was fabricated using 3D printing and laser cutting, in PETG and Acrylic respectively. A HC-06 Bluetooth module was used to receive Bluetooth commands from a custom Android controller app, designed in MIT App inventor 2. By considering the torque generated by the servo motors, the arm was able to pick and place objects up to 80g in weight within its operational range while drawing up to 970mA of current. Several tests were conducted to study the system’s performance in terms of lifting capacity, connection range, idle current draw, and reliability. The system was able to achieve real time operations and be manually controlled or be set to run a predefined algorithm.

The robotic arm could be redesigned to allow more efficient transfer of torque from the servo motors to the limbs using a wider application of mechanical levers. This would allow a greater force to be generated on the shorter length of a lever by using a fulcrum, increasing the load that the system can move with the trade-off of losing range of motion. A new detection algorithm can also be developed to enable the system to detect objects within range of the end-effector and allow the automatic reaction for pick and place operations. For example, a Time of Flight (TOF) module could be installed to allow the system to detect how far away an object is and automatically pick it up when at a predefined distance.

REFERENCES

Benchmark of Sampling Based Motion Planners in Bin Picking Manipulation Task

1st Sebastian Realpe  
Dept. of Mechatronics Engineering  
Universidad Nacional de Colombia  
Bogotá, Colombia  
srealper@unal.edu.co

2nd Felipe Gonzalez Roldan  
Dept. of Mechatronics Engineering  
Universidad Nacional de Colombia  
Bogotá, Colombia  
fgonalezro@unal.edu.co

3rd Jose Manuel Fajardo  
Dept. of Mechatronics Engineering  
Universidad Nacional de Colombia  
Bogotá, Colombia  
jmfajardod@unal.edu.co

4th Juan D. Hernández  
School of Computer Science and Informatics  
Cardiff University  
Cardiff, Wales-UK  
HernandezVegaJ@cardiff.ac.uk

5th Pedro-F. Cardenas  
Dept. of Mechatronics Engineering  
Universidad Nacional de Colombia  
Bogotá, Colombia  
pfcardenash@unal.edu.co

Abstract—There is a wide variety of motion planning techniques for robots; however, there are few tools to test and compare the performance of motion planners in various environments. This work proposes a test layout for a set of industrial robots to measure their performance in manipulation in a constrained environment. The developed work uses ROS and the benchmarking tool Robowflex. The comparison allowed to obtain the performance results of multiple motion planners in a shorter time than in previous works. These results facilitate the correct choice of a motion planner, which is critical in current industrial environments. The conducted benchmark provides a valuable tool for potential comparisons of new motion strategies to proven and established methods.

Index Terms—Robotics, Motion planning, ROS, Benchmark.

I. INTRODUCTION

One of the critical research elements in robotics, specifically in the field of manipulators or robot arms, is resolving questions such as how will the robot move? Which path is it going to follow? Which joint values are needed to achieve a given task? Usually with some algebra background and some Robotics 101 theory, the problem can be solved simply by calculating the inverse kinematics of the robot. For some cases, this may be enough, but, in most robot applications, there are obstacles involved in between; what if point A is in a box and point B is in a shelf? The problem sounds less trivial now than just employing inverse kinematics. Here is where motion planning comes into play.

Motion planners make it possible for a robotic system to plan ahead on how it will move through an environment by avoiding obstacles, self-collisions, and collisions with the robot tool or object in the robot tip. Some of these not only solve the problem of obstacles but also aim to improve time, path length, and path smoothness, because, if possible, the robot will get to the requested goal, but it is also essential to plan how it is required to get there.

Multiple techniques and algorithms are used for motion planning, such as artificial potential fields [1], graph search-based algorithms like D-star [2], or optimization-based as Covariant Hamiltonian Optimization for Motion Planning (CHOMP) [3]. This work uses some of the motion planners available in the Open Motion Planning Library (OMPL) [4] in a robot arm context. OMPL holds a collection of motion planners known as sampling-based motion planners (SBMP). SBMP generate random samples of the robot joint states in its configuration space to form a path by connecting these states, as long as they are valid [4]. Even though SBMP share this characteristic, each planner in OMPL is intended to solve sub-tasks as collision avoidance, shortest path calculation, computation time minimization, and others. Different approaches imply having pros and cons; some methods could be slower than others, provide longer paths, or not even be able to compute a feasible path in a given time.\footnote{Most sampling-based algorithms are categorized as probabilistically complete, this means that if a solution exists, the algorithm will find it if given enough time. [4]}

These situations can be highly related to the specific task the robot is performing, the robot morphology, or the working scenario.

This paper aims to produce a benchmark between the OMPL implementations of the motion planners RRT, RRTConnect, PRM, LazyPRM, KPIECE, BKPIECE, LBKPIECE, and EST, in a representative scenario of a bin-picking task for a robot manipulator. The selection of the motion planners lies in their wide use in other benchmarks [5], [6], as well as in their popularity for measuring the performance of new planning techniques for robots [7]. The experiment consists of using three different simulated robots. Each planner will attempt to find a path for the proposed task; they will be run multiple times in each robot to collect data and obtain statistics of computing time and successfully solved runs.
These parameters accurately describe the motion planner’s performance in the given scenario.

This work is organized as follows: Firstly an short introductions is presented, then this paper continues with the Background in section II where we talk about the Related Work (II-A) to this work, the Software (II-B) used, and a brief description of the Motion Planners (II-C) benchmarked. Next, the Problem Formulation on section III the employed Manipulators (III-A) models are specified, the proposed task Scenario (III-B) is described and the Performance metrics and Benchmark parameters (III-C) used are defined. Then the Results in section IV where the post-processed data related to the metrics discussed previously is exposed to allow a statistical analysis. Based on this data an analysis of the results is taken in the Discussion section V. Finally, in section VI the final comments regarding the overall topics of this paper are presented.

II. BACKGROUND

Benchmarking allows comparing and evaluating different approaches to accomplish a common goal. In the context of this work, benchmarking is useful for comparing motion planners when given a motion task or request.

A. Related work

Previous works have discussed results of using OMPL planners in different environments and tasks. J. Meijer et al. proposed a benchmark on three robots to evaluate the performance of the planners in grasp-related tasks, having robots working in somewhat constrained workspaces [8]. C. Wang et al. proposed a method for motion planning based on Trajectory Splitting and performed a benchmark with OMPL planners to compare its performance [9].

Due to the diversity of robots and their tasks, there will always be a niche for benchmarking tools. Therefore, it is essential to have fast and reliable tools to test motion algorithms and compare their results with established and functional techniques.

In this scenario, the work developed is linked to un-updated software and slow simulation tools. Hence our work tackles both fields, incorporating RobowFlex as a benchmarking tool and developing all the layouts in Robot Operating System (ROS).

B. Software

OMPL offers an Application Programming Interface (API) to use these motion planners in several frameworks and applications easily. One of the frameworks where OMPL is widely used is MoveIt, a ROS-based open-source tool that enables users to deploy and perform basic motion queries for robot manipulators quickly [10]. A Setup Assistant Graphical User Interface (GUI) allows users to generate the initial setup and motion configuration files for the desired manipulator. It also provides an API to use its functionalities through C++ or Python code.

MoveIt serves as a middleware, managing the layers of scene configuration, motion planning, collision checking, and motion execution. While it helps to integrate these layers as a whole, it also makes it difficult to access them separately, e.g.: to modify parameters on the run, add or edit planning strategies, or obtain specific planner information data.

ROS was used to define the simulated robots in Unified Robot Description Format (URDF), and MoveIt generates the robot’s configuration package, which sets up the benchmark scene and the motion query. For benchmarking, we used RobowFlex [11], a tool that aims to overcome some of the MoveIt limitations, particularly the issues related to the usage and changes of the motion planning layer; in our case, for extracting planner data in a motion request. Robowflex is then a useful tool for this work because it allows a quickly developing of the setup and execution of the benchmark experiment with few lines of code. Example 1 is a portion of code 2 corresponding to a benchmark setup and execution.

```c++
// Setup a benchmarking request
Profiler::Options options;
options.metrics = Profiler::CORRECT;
Experiment exp("exp_name", options, 5.0, 100);
// Create a motion planning request
auto request = std::make_shared<MotionRequestBuilder>(planner, GROUP);
request->fromYAMLFile(".../request.yaml");
request->setConfig("planner_name");
//Setup experiment
experiment.addQuery("planner_name", scene, planner,
request->getRequest());
//Execute benchmark
auto dataset = experiment.benchmark(4);
OMPLPlanDataSetOutputter output("bench_results");
output.dump(*dataset);
```

Example code 1: Example of a experiment setup using Robowflex.

Robowflex can record the benchmark results and save them into a log file, this log file can be parsed to a database structure using a MoveIt Python script available in the benchmarking MoveIt Tutorial page, this database file can then be uploaded to the MoveIt Planner Arena [12] application, which processes the data and offers various forms of visualization for it. It can also be used by external applications compatible with SQL queries to handle the data as required, as we used Python to execute the posterior statistical analysis and also the generation of the upcoming graphs in the Results section (IV).

C. Motion planners

The OMPL algorithms evaluated in this paper are single query motion planners, this means that for each motion request, the algorithm generates from the ground up the structure that defines the specific trajectory for the request.

1 Fragment of one of the examples available in the RobowFlex library resources. The code used for this paper is based on a benchmark example.
In contrast, multi-query planners use the same generated structure, or part of it, to calculate a motion request when the task environment has not changed much or not even at all.

A brief overview of the motion planner algorithms benchmarked in this paper is presented for a close understanding of their core behavior:

1) **RRT**: It stands for Rapidly-exploring Random Tree [13]. It is both a data structure and method designed for path planning problems. The tree structure has its root in the start state of the query, from where branches start growing towards randomly sampled nodes in the free configuration space. A branch consists of two nodes or vertices connected by an edge, and they grow until it reaches a minimum distance to the goal state. Although these nodes are random, they are biased to be placed in the least explored sections of the configuration space and validated to be collision-free. The closest node to another will grow a branch towards it.

2) **RRTConnect**: This method [14] builds two RRTs with roots at the start and goal state. Both trees start their behaviors as RRTs, growing in the free space towards each other. When the nodes of both trees meet at a minimum distance, the trees are connected, creating a path for the motion request. This behavior improves convergence speed over the original Rapidly-exploring Random Tree (RRT) algorithm.

3) **PRM**: The Probabilistic Roadmap algorithm [15] first constructs a roadmap of nodes in the robot's configuration space. Each node is validated to be in a collision-free state configuration of the robot, and the edges connecting them are a feasible path between them. When the construction is done, the roadmap is stored as a graph. The motion request now becomes a graph-based search problem, where the objective is to find a path in the roadmap that joins the Start state with the Goal state using graph search algorithms such as A-star.

4) **LazyPRM**: This algorithm [16] follows the same procedure as PRM, but it is assumed that the nodes are in a collision-free state of the roadmap in the construction phase. Once the graph is built, the shortest path between the start and goal states is sought. Each node and edge of this path is validated to be collision-free, if this is not the case, the algorithm removes the node from the roadmap, and the search for a path in the roadmap starts again until a collision-free path is found.

5) **KPIECE-Family**: The KPIECE algorithm (Kinodynamic Planning by Interior-Exterior Cell Exploration) was proposed by Sucan et al. in [17]. KPIECE builds a tree in the configuration state space of the robot. From an initial configuration, a multiple-level grid-based discretization guides the tree transitions to the less explored regions of the search space. This technique is intended for complex dynamics systems, consequently, a physics simulator is required to compute the transitions between states. BKPIECE stands for Bidirectional-KPIECE, it uses the concept of bi-directional trees [14], in which two trees are created: one from the initial configuration, and another from the goal configuration. The exploration process follows the same principle as the KPIECE but tries to connect both trees. Finally, the LBKPIECE incorporates the principle of lazy collision evaluation [16] into the BKPIECE.

6) **EST**: The Expansion-Space Tree [18] is a single query planner that covers the state space by trying to connect an initial configuration and a goal configuration using a tree. A local planner attempts to connect random samples generated by a probabilistic function. The local planner searches for feasible paths between the sample and the tree nodes: if the path exists, the shortest one is selected, and the sample is added to the tree, which keeps growing until the goal configuration is reached.

It is worth noting that in sampling based planners, a sample corresponds to a set of joint values (configuration-space $\{C\}$), which represents a pose in the task space.

### III. Methods

The benchmark was carried out in a computer equipped with an Intel core i7 6700 processor at 2.4 GHz, 4 cores/8 threads, 16 GB of RAM, and an Nvidia M1000M graphics card. The experiment was performed using ROS Noetic on Linux 20.04, Robowflex v1.3, Moveit 1.1.5 Alpha, and OMPL v1.4.2.

#### A. Manipulators

Although sampling-based algorithms rely on randomness for the structure generation, it is interesting to evaluate if they follow a tendency in similar conditions but with some variations. For this purpose, the proposed experiment was conducted on three different simulated robots of industrial grade: the IRB140 from ABB Robotics [19], the KR6-900 from KUKA [20], and the UR5 from Universal Robots [21]. These robots share familiar features such as 6 degrees of freedom, a similar shape profile, and the ability to perform the same task with the same start and goal states.

The robot tool used to simulate the bin-picking task is a simple two-finger claw-like gripper designed in and for the Laboratory of Smart Robotic Systems (LabSIR) at Universidad Nacional de Colombia. This gripper adds about 20 cm in length to the tip of each robot.

#### B. Scenario

The proposed scenario is a bin-picking task, where the starting configuration of the robot is towards its left side, inside a bin where several objects are supposed to be, as seen in Fig. 1. The goal position is on a shelf directly in front of the robot. This is a common task scenario for a robot manipulator, it has been employed in academic challenges such as the Amazon Picking Challenge [22] and in industrial applications such as [23] or [24]. Motion planners must determine a feasible path avoiding collisions with the bin walls and the shelf structure.

#### C. Performance metrics and benchmark parameters

The following metrics are used to study the performance, in the given task, of the benchmarked motion planners. These metrics are related to the benchmark setup options, planning timeout, and total planner runs.
1) **Planning time:** Given the probabilistic completeness of SBMP, the more time given to a planner to compute a path increases the probability to find a solution [25]. However, the planning time affects the overall time of the task, so the faster an algorithm can come up with a solution, the better. Measuring the time that a planner takes to find a solution also measures its performance.

A timeout end-condition ensures that the planner does not take too long trying to find a solution, if this timeout is reached, another attempt can be made, so the planner restarts the motion query to find a path. For this benchmark, a timeout of 5 seconds was the limit set for all planners.

2) **Success rate:** The second metric is the success rate for solving a motion request. This metric is defined as the number of times the planner has successfully found a path within the given time limit. This metric shows the reliability of a planner for a request in the given time window.

To generate the data for this metric, each algorithm was run 100 times, with the same parameters, start and goal states, and scene objects. Counting the successful runs provides a direct value of the success rate.

3) **Planner specific parameters:** Each motion planner has specific parameter values that control the behavior of the algorithms. For this benchmark, the default OMPL parameter values were used; these are as stated in Table I.

### IV. RESULTS

The results obtained in the benchmark show aspects such as the success rate, planning computation time, and key aspects of motion planners performance.
V. DISCUSSION

A. RRTConnect and LazyPRM

The RRTConnect and LazyPRM algorithms show an outstanding performance in all three queries of the simulated robots with 100% success rate and the lowest computation time with minimal deviation. These algorithms are derived from RRT and PRM. The improvements regarding path generation and collision checking respectively can be verified in terms of computation time by comparing the overall performance with RRT and PRM.

B. RRT

The RRT planner shows the poorest performance among all the robots for the given experiment, especially in the case of the KR6 and the UR5 robots, as it often reached the timeout limit, which also means frequent unsuccessful runs. This result places the RRT as the least reliable planner for the specific task proposed in this paper. Variables like shelf position, bin size, robot position, and size, among other parameters involved in the experiment have an effect on the space exploration performed by this algorithm, thus making the configuration space hard to traverse based purely on random sampling.

However, it is important to remember that RRT is still a probabilistically complete planner, so we conducted an additional isolated experiment to explore what would the RRT planner achieve for the KR6 and UR5 robots with more time to find a path. The test was set to 100 runs and a timeout of 600 seconds, Fig. 3 presents the obtained results.

![Fig. 3: Isolated test for the RRT with 600 seconds of timeout on the KR6 and UR5 robots.](image)

Although the UR5 robot took an average of 140 seconds, the box plot shows a somewhat concentrated behavior around this average. The opposite happens with the KR6 robot, where the total computation time to find a path by the RRT takes the whole range, from as little as 10 seconds to close to the maximum time of 600 seconds, which illustrates of how sparse could be the sampling and tree growth behavior of this algorithm.

C. KPIECE Family

Understanding the KPIECE family as the set of planners based on interior/exterior cell exploration (KPIECE, BKPIECE, LBKPIECE), they exhibit a varying behavior regarding computation time for each robot, e.g.: the UR5 obtained the least average time for the KPIECE, but the maximum in both BKPIECE and LBKPIECE. Although with the UR5 robot, the LBKPIECE only achieved 41% of success, the remaining success rate of the family for all robots is 100%, which gives them a reasonable level of reliability.

D. PRM and EST

The PRM and the EST occupy second place in sampling-based motion planners with the best performance. Both planners have a 100% success rate and, in terms of computation time, a pretty average value in mean and standard deviation in comparison with the other planners.
E. Robowflex vs Moveit Benchmark Tool

Initially, the benchmark was conducted using entirely Moveit assets, the application offers its own benchmarking pipeline and procedures [26]. At the same time, we found out about Robowflex [11] by a fellow researcher at the Kavraki Lab. at Rice University. Doing tests, the benchmark using the Moveit framework experiment for one robot took 17 minutes, while one conducted with Robowflex only took 2 minutes. As Robowflex proved to be highly optimized for benchmarking these motion planners, we opted for using it as the principal software tool for the experiments.

VI. CONCLUSIONS

This paper presents a successfully executed benchmark of motion planners in a pick and place scenario using an efficient software known as Robowflex. Crucial aspects of the benchmark of motion planners were presented such as software used, planner performance, and task configuration.

The software involved played a significant role by using recent tools such as Robowflex to efficiently code, execute the experiment, and integrate easily the OMPL implementations of various SBMP. Also, Moveit as an application to set up and save the robot’s environment and motion request, both as YAML files readable by Robowflex. The previously mentioned software is used under the ROS framework.

Regarding the obtained results in this benchmark, it is clear that planners like RRT and PRM are outperformed in the three cases(robots) by their improved counterparts, the RRTConnect and the LazyPRM, concerning the computation time. For future work, it is ideal to propose different scenarios or tasks to evaluate if this performance is maintained or is a matter of specific cases and configurations. As for the KPIECE planner family, in Fig. 2 is unclear if there is any improvement between them. For example, the KR6 robot shows better results with LBKPIECE and BKPIECE than with the KPIECE, but the opposite happens with the IRB140 robot, where the best result is yielded by the KPIECE planner.

In all cases, planner performance is affected by several variables, starting with the parameters that modify their behavior; but it remains unclear how to practically tune them. Another variable is the task to accomplish, which involves the request and the environment where it is executed. It is difficult to propose an optimal scenario with obstacles, because an easy-to-traverse task space may be a maze in the C-Space. Lastly, robot morphology also impacts planner performance. In summary, achieving an outstanding planner performance is a matter of fine-tuning of parameters and intelligent choice of the motion planner.

REFERENCES


Tether-Based Localisation System for Underwater Robots

A. Martin and Dr. G. Tang
School of Aerospace, Transport and Manufacturing, Cranfield University
Cranfield, Bedford, UK
achillemartinpro+cranfield.uni@googlemail.com, g.tang@cranfield.ac.uk

Abstract — Tank-inspection robotic crawlers often rely on localisation techniques such as dead reckoning to survey walls and surfaces. However, the robot position can drift over time when using these techniques, which therefore require precise calibration. This paper presents a novel tether-based system providing a real-time localisation reference tool to help calibrate and understand prevalent localisation technologies, inside uncluttered tanks. The assessed localisation technologies can subsequently be used in cluttered and complex operational tanks. The developed system relies on a mechanical connection between the tank edge and the robotic crawler and is composed of two distinct units. The first unit is the tether orienting platform clamped onto the tank edge and guiding a tether coming out of a draw-wire position sensor. The second unit is the crawler platform mounted on top of the robotic crawler and holding the tip of the tether, while allowing the robot to move freely. The performance of the system is assessed by determining the accuracy on the 3D position of the robot and the accuracy on its 2D heading, over a 1m x 1m area. The positional accuracy with 90% confidence interval is ±39mm and the heading accuracy with 95% confidence interval is ±4.8°.

I. INTRODUCTION

A. Background

Real-time localisation is crucial for remote controlled systems as well as autonomous systems especially in cluttered [1], hazardous [2], and unknown environments [3]. On the one hand, it enables the operator to remotely track the location of the robot, on the other hand, it enables autonomous systems to constantly receive an estimation of their position and orientation and take actions accordingly. Localisation is also important for survey or inspection tasks, which rely on the mapping and understanding of the environment, to correlate observations from the robot with their precise localisation. Thus, developing accurate and affordable localisation technologies and algorithms is vital for the robotic field related to (underwater) exploration.

B. Literature review

State-of-the-art localisation systems for underwater robots include sonar-based and tether-based technologies. The sonar-based technology originates from the Neptune system (North-East Pacific Time-Series Undersea Networked Experiments), an underwater robot used to remotely inspect aboveground storage tanks [4]. The next generation of its acoustic localisation system (Sonic High Accuracy Ranging and Positioning System II known as SHARPS II) relies on high-frequency, spread-spectrum techniques to produce precise time-of-flight range measurement with sub-centimetre scale resolution [5].

The SHARPS technology is similar to baseline acoustic positioning systems which include long baseline (LBL), short baseline (SBL) and ultra-short baseline (USBL) systems. LBL systems measure ranges between a shipboard hydrophone and two or more transponders mounted at the bottom of the ocean [6]. SBL systems require three or more shipboard hydrophones receiving data from a single beacon at the bottom of the ocean. USBL systems can provide accurate measurements with a single multielement shipboard hydrophone and a single beacon on the ocean bottom. Each baseline system can be used for a specific application and environment. According to [5], [7], and [8], their scale resolution is on the order of metres and their heading accuracy is less than a degree.

Sonar-based technologies are accurate but expensive to install and maintain. In contrast, tether-based technologies are relatively low-cost for underwater real-time localisation. The reference for tether-based underwater localisation systems is the Smart Tether, patented by US company KCF Technologies in 2013 [9]. This tether is composed of sensors embedded within a portion of a flexible tether. Knowing the spacing between each sensor along the cable, and using the 2D catenary curve equations consecutively, the position of each sensor can be determined in the 3D space, as explained in [9]. The product is still being sold in 2022 by VideoRay and shows good performance for small/medium ranges (40m) with a scale resolution on the order of metres and a heading accuracy of less than a degree.

The tether-based technology is more practical for punctual underwater investigations, and although a bit less precise, it is more affordable than the LBL. Improved tether-based technologies relying on fibre-optic shape sensing (FOSS) [10] are more accurate. These technologies can involve the use of Fibre Bragg gratings (FBG) [11], to provide discrete bending parameters of the tether, and combined with 3D curve equations, help derive the shape of the tether [12]. Although they are light, accurate, geometrically versatile, they lead to complex implementation in harsh environments (e.g. water environment) where they are fragile, sensitive and intended at small-scale use.
Tether-based technologies can also be found in the form of tether length and orientation determination systems rather than tether deformation systems. The leading device able to determine the tether length is called a draw-wire position sensor, or cable-extension transducer system (CET), presented in [13]. Pulleys are combined with the CET and relying on a pan-tilt mechanism to ensure that the cables come out with the same orientation as the robot. A triangulation algorithm is then used to compute the position of a second robot attached to a master robot, using a paired CET system. Another tether-based technology used to evaluate the orientation of an object in 3 dimensions is presented in [14]. The position control concept includes a cable-suspended platform, encoder systems (force sensors, potentiometers) and cable length measurement systems. The performance of a custom combined position and orientation determination system can reach a scale resolution on the order of millimetres for a range of around 10m and a heading accuracy of less than a degree.

Alternative underwater localisation technologies exist but face significant drawbacks such as GPS systems being unreliable under water and camera-based systems not being performant in poorly lit environments.

C. Problem definition

Nautilus bathyscaphic robot is a tethered underwater robotic crawler funded by Innovate UK for storage tank inspection, initially relying on dead reckoning for localisation. Over long distances or long operating time, the position error with dead reckoning method increases gradually. For instance, a bias error in acceleration becomes linear in velocity and quadratic in position. A solution to alleviate the error drift is to rely on another localisation reference or calibration system to update the position and orientation of the robot after a specific duration, or to compensate the error increase thanks to a thorough calibration process.

The aim of the paper is to design and create a system for calibrating and understanding dead reckoning localisation estimation errors. The system must provide accurate 3D position and 2D heading orientation estimations of the centre of mass of the robotic crawler in real-time. The positional accuracy of the prototype must achieve ±20cm over a 5m range (diameter of the test tank) and its orientation accuracy must achieve at least ±5° to provide exploitable data for the localisation technologies to be correctly calibrated. The uncertainties provided are given according to a 90-95% confidence interval.

II. METHODOLOGY

A. Theoretical localisation estimation

In order to get a theoretical estimation of the positional and orientation accuracy of the proposed system, an algorithm was created. MATLAB was selected to handle complex matrix calculations. Relying on a forward kinematics approach to derive the position of the robot from random “joint configurations” (representing configurations of the system), coupled with error propagation formulas to take into account the individual error of the various sensors, the algorithm averaged the estimated standard deviation of the position and orientation error for 1000 random realistic configurations. A 1m-long wire was considered for the theoretical model to serve as reference for the experiment.

B. Experimental localisation estimation

To assess the positional and orientation accuracy of the assembled innovative localisation system, the OptiTrack by NaturalPoint Inc was utilised. The OptiTrack provides precise 3D tracking of any entity thanks to physical markers, high-end cameras, and the Motive software suite.

Three cameras were selected to monitor a 1m x 1m area representing the bottom of the tank. The setup, in the aerial environment to ensure good OptiTrack performance, is shown in Fig. 1. Once calibrated, the OptiTrack can produce positional error less than 0.3mm and rotational error less than 0.05°, according to the manufacturer.

Five motion capture markers have been placed directly onto the robotic crawler to determine the position of its centre of mass in 3D in real-time, as well as the heading orientation of the rigid body. Using five markers instead of three ensures that at least three markers are visible at all times by the OptiTrack and that the measurements are uninterrupted if one or two markers are hidden by the body, for instance one marker is hidden in Fig. 1. Four markers have also been placed onto the tank bracket holding the innovative system to set the origin of the work area and define a reference frame for the robot motion.

To compute the estimated position and orientation of the centre of mass of Nautilus in real-time from sensor measurements, an Arduino MEGA 2560 has been selected for its number of interrupt pins, performant Analog-to-Digital converter (>10-bit resolution), small size (maximum 100 mm * 100 mm * 50mm) and affordable price. The MATLAB localisation estimation program used in the simulation phase has been converted into an Arduino program to make use of real-time sensor data and has been run on a laptop (Huawei Matebook D14). Nautilus was driven around the free space, while the Arduino board processed the data coming from the

Figure 1. OptiTrack cameras initial setup
sensors and sent the position and orientation estimation of the crawler to MATLAB main program (3Hz updates).

In parallel, Motive software was recording, via the OptiTrack cameras, the position and orientation of Nautilus centre of mass in real-time. The OptiTrack data was sent to MATLAB with a user-defined delay every time the Arduino program collected data from system sensors. This way, measurements from different sources were taken at the same time i.e. representing the same position and orientation of the robot while it was moving. The MATLAB program then gathered the position and orientation estimations into a unique table. The data acquisition structure is shown in Fig. 2.

During the data analysis phase, outliers representing OptiTrack data miss (less than 3 visible markers during capture time) or a sudden shift in values (external disturbances such as lighting), were manually removed from the table.

For the experiment, Nautilus was driven randomly across its free 2-dimensional space, at a specific floor height, for 5 minutes, which corresponds to 800 different configurations recorded after outlier removal.

III. RESULTS AND DISCUSSION

A. Technical details

To comply with the main requirements from the calibration project (uncluttered test tank with a height of 1m, a diameter of 5m and a flat bottom), and according to the literature review, the most suited technology for the localisation system is a tether-based technology. A mechanical connection between the tank edge and the robotic crawler provides various benefits such as: satisfying accuracy, easy setup, versatility, affordability.

In order to evaluate the position and orientation of the robot inside the tank, the system architecture presented in Fig. 3 has been adopted.

The localisation problem has been solved with a single-wire localisation approach inspired from tethered simultaneous localisation and mapping (TSLAM) [15], relying on the determination of 2 angles (horizontal and vertical rotations) and tether length, as shown in Fig. 4.

The innovative system used to determine the 3D position and 2D orientation of the underwater crawler can be divided into two subsystems or units.

The first subsystem is a rotating draw-wire platform, placed on top of a tank attachment entity, which includes a deflection pulley and wire-guiding system to ensure that the desired angles are precisely measured and that the mechanical disturbances are minimised. The angles are parametrised by $\theta$ and $\lambda$ shown in Fig. 4.

The second subsystem is the crawler attachment used to connect the wire to the robot while providing the heading angle of the rigid body (angle not represented in Fig. 4).

The assembled subsystems, combined with 1 draw-wire position sensor (AK Industries CD50 1000-R01K-L15-K01-OP-IX), 3 rotary encoders (Wisamic encoder LPD3806-600BM-G5-24C), is presented in Fig. 6.

B. Theoretical results

Figure 2. Real-time data acquisition structure for system testing

Figure 3. Subsystem architecture and interfaces

Figure 4. Side and top view for the single-wire localisation approach

Figure 5. Forward kinematics explanatory model of the system
The theoretical model has been approximated with the model shown in Fig. 5 and used to derive the Denavit-Hartenberg (D-H) parameters of the forward kinematics model.

To determine the accuracy of the system and compare it to other existing systems, the standard deviation is calculated using the error propagation rules. The standard deviation for each coordinate of the position of the crawler (\(\sigma_x, \sigma_y, \sigma_z\)), as well as the standard deviation for the heading angle of the crawler (\(\sigma_{\text{heading}}\)), are calculated from the standard deviations of the measuring devices and production devices (3D printing machine).

The error propagation formula is calculated for any variable \(Z\) which depends on other variables according to the function \(Z = f(A,B,C)\). The uncertainty on \(Z\) (\(\Delta Z\)) can be expressed with the uncertainties on the other variables \(A, B, C\), as shown in (1).

\[
\Delta Z^2 = \left(\frac{\partial f(A,B,C)}{\partial A}\right)^2 \sigma_A^2 + \left(\frac{\partial f(A,B,C)}{\partial B}\right)^2 \sigma_B^2 + \left(\frac{\partial f(A,B,C)}{\partial C}\right)^2 \sigma_C^2
\]

\(\Delta Z\) represents the uncertainty with 68% confidence interval. To obtain an estimation with 95% confidence interval, \(\Delta Z\) has to be doubled: \(\Delta Z(95\%) = 2 \times \Delta Z(68\%)\).

Uncertainties with 68% confidence interval can be determined from different sources: the manufacturer providing the resolution or accuracy of the device, a single measurement with the device (uncertainty type B) or a sample study (uncertainty type A).

The propagated errors for the variables of interest (position coordinates and heading angle) can be derived from the uncertainties with 68% confidence interval for each device used and from formula (1). The values are shown in Table I.

According to [16], the Mean Radial Spherical Error (MRSE) can be calculated to determine the 3D accuracy of the position. It represents the radius of sphere centred at the true position, containing the position estimate in 3D with probability of 61%. To obtain the radius of sphere centred at the true position, containing the position estimate in 3D with probability of 90%, the formula shown in (2) is used.

\[
\text{MRSE}(90\%) = 0.833 \times (\sigma_x + \sigma_y + \sigma_z) \quad (2)
\]

Uncertainties with 90-95% confidence interval are subsequently calculated and shown in Table I.

<table>
<thead>
<tr>
<th>TABLE I.</th>
<th>POSITION AND HEADING UNCERTAINTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncertainty</td>
<td>Value</td>
</tr>
<tr>
<td>MRSE(90%)</td>
<td>4.5mm</td>
</tr>
<tr>
<td>(\sigma_{\text{heading}}) (95%) = 2 \times \sigma_{\text{heading}} (68%)</td>
<td>0.26°</td>
</tr>
</tbody>
</table>

C. Experimental results

The prototype of the innovative system used for the experiments is shown in Fig. 6.

The 2D point cloud representing the successive positions of Nautilus is shown in Fig. 7. The point (0 mm, 0 mm) in Fig. 7, denoted by a red asterisk, represents the origin of the localisation system reference frame and is located at the centre of the tank edge bracket.

From the data collected by the various system sensors after outlier removal, standard deviations of the position and orientation errors, in all directions (X, Y and Z), are calculated and gathered in Table II. The mean calculated for each error (difference between the estimations from the innovative system and from OptiTrack) is near 0, but a slight bias persists: \(X_{\text{bias}} = 36\text{mm}, Y_{\text{bias}} = 48\text{mm}, Z_{\text{bias}} = 31\text{mm}\). This means that there is a slight offset between the system reference frame and OptiTrack reference frame.

<table>
<thead>
<tr>
<th>TABLE II.</th>
<th>COMPARISON OF SYSTEM UNCERTAINTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard deviation (68%)</td>
<td>Localisation system (simulation – 1m range)</td>
</tr>
<tr>
<td>(\sigma_x)</td>
<td>19mm</td>
</tr>
<tr>
<td>(\sigma_y)</td>
<td>23mm</td>
</tr>
<tr>
<td>(\sigma_z)</td>
<td>4.7mm</td>
</tr>
<tr>
<td>(\sigma_{\text{heading}})</td>
<td>2.4°</td>
</tr>
</tbody>
</table>

The standard deviation \(\sigma_z\) presented in Table II is calculated for information and used to derive a positional accuracy which can be compared with literature. Since the
experiment has been run for a constant height, the position error only corresponds to a specific height.

According to Table II, there is a clear difference between the experimental results and the simulation results. The higher standard deviation for the prototype can be caused by inaccuracies in OptiTrack marker positioning, a misalignment during the incremental encoder calibration or fluctuating mechanical constraints on 3D printed material, resulting in part bending.

D. Comparison with literature

1) Experimental results

To assess the degree of accuracy achieved by the experimental localisation system, position and orientation uncertainties have been calculated to compare with results found in literature. The positional accuracy and orientation accuracy have been calculated for the experimental localisation system and are compared with state-of-the-art localisation technologies in Table III.

<table>
<thead>
<tr>
<th>System</th>
<th>Range</th>
<th>Position accuracy (90% uncertainty)</th>
<th>Orientation accuracy (95% uncertainty)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Localisation</td>
<td>1m</td>
<td>39mm</td>
<td>4.8°</td>
</tr>
<tr>
<td>system(Exp)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation</td>
<td>1m</td>
<td>4.5mm</td>
<td>0.26°</td>
</tr>
<tr>
<td>Simulation</td>
<td>5m</td>
<td>18mm</td>
<td>0.26°</td>
</tr>
<tr>
<td>Smart Tether</td>
<td>40m</td>
<td>1.5m</td>
<td>0.5°</td>
</tr>
<tr>
<td>LBL</td>
<td>100m</td>
<td>0.2m</td>
<td>0.1°</td>
</tr>
</tbody>
</table>

According to Table III, the positional accuracy obtained for the localisation system is better than the one obtained for the KCF Smart Tether or the LBL. However, the orientation accuracy is 10 to 50 times higher. This highlights the fact that there is a significant factor perturbing the localisation system, or that the localisation system technology is not suited for orientation estimations. This issue seems to originate from the prototype, as the MATLAB simulation can provide an orientation accuracy better than the Smart Tether.

When, as a second step, the measurement range is taken into account, it is more convenient to compare the ratio accuracy over range. The comparison between ratios of accuracy over range is displayed in Fig. 8.

According to Fig. 8, the prototype provides a positional accuracy which can compete with the accuracy of the Smart Tether. However, it is outperformed by the LBL, which displays an even better ratio than the simulated localisation system. The latter comparison proves that with the current components and their characteristics, it is impossible, even ideally, to beat the LBL results. The performance of the localisation system can be enhanced, but at a higher cost.

Concerning the orientation accuracy ratio comparison displayed in Fig. 8, the prototype is far behind the other systems. Even the simulation designed for the project (5m range) cannot achieve a better orientation accuracy/range ratio than the Smart Tether. The unsatisfactory orientation accuracy confirms that the localisation system is intrinsically not optimised for accurate orientation estimations.

2) Enhanced model

At this stage, the comparison between the experimental results and the literature is not fully relevant, as the localisation system can only be used in the aerial environment and not underwater. Switching one Wisamic encoder with an IP68-compliant encoder might enhance the accuracy but will also influence the cost. The performance comparison is presented in Fig. 8.

Different enhancements have been suggested for the 5m measurement range:

- Underwater model: the rotary encoder on the crawler platform is IP68-compliant.
- Robust model: Sheet metal is used for the mechanical architecture.
- Draw-wire+ model: more performant draw-wire.
- Encoder+ model: more performant rotary encoder.
- Underwater, robust model+: all of the above.

According to Fig. 8, the robust submersible model for 5m measurement range performs better than the LBL and is 300 times less expensive. The simulation results confirm that the developed tether technology is viable for accurate position estimations. This model also outperforms the Smart Tether and is 5 times less expensive. However, the tether technology relying on
rotary encoders is outperformed by the performance of the compass, according to the orientation ratio results.

An improved mechanical architecture will significantly increase the positional accuracy without major increase in cost, but it will have limited impact on the orientation accuracy. A better draw-wire will increase cost without significantly enhancing the positional accuracy or the orientation accuracy. Better encoders will have a positive influence on the positional accuracy, but most importantly, will lead to an orientation accuracy gain, while having a moderate impact on the cost.

IV. CONCLUSION

An accurate tether-based localisation system relying on tether technology has been proposed and implemented in this work to serve as a reference tool for other localisation technologies, such as dead reckoning or sonar-based technologies. A prototype has been produced and its positional accuracy and orientation accuracy have been assessed by an extremely accurate and vision-based motion capture system called OptiTrack.

The prototype has been able to determine in real-time (3Hz updates) the position of the centre of mass of Nautilus (robotic crawler), with an uncertainty of ± 39mm over a 1m measurement range (90% confidence interval). It has also been able to determine the heading orientation of Nautilus with an uncertainty of ± 4.8° (95% confidence interval). The prototype has been identified as competitive against other tether-based state-of-the-art solutions such as the Smart Tether by KCF in terms of positional accuracy and cost.

Increasing the rigidity of the mechanical device combined with high performance rotary encoders and draw-wire position sensor, the positional accuracy of the new system could outperform the results from the LBL, while remaining affordable for industrial companies. However, it has been underlined that this enhanced system complying with the research objectives could not outclass the compass, used jointly with tether-based and sonar-based technologies in terms of orientation accuracy. Nevertheless, the enhanced system has been validated to serve as a reference tool for other positioning technologies, as it performs identically or better than the current best solutions on the market and is cost-competitive.

Another method to increase the accuracy would be to implement supervised learning and involve training an AI model using the real orientation and position data from OptiTrack sensors and correlate it with the data coming out of the innovative system sensors.

ACKNOWLEDGEMENT

Thank you to InnotecUK for their support on this project and in particular to Tim Neat, Mechanical Lead Engineer, for his extensive knowledge and creativity.

REFERENCES


Collision-free Path Planning For Welding Manipulator Via Deep Reinforcement Learning

Bo Hu, Tao Wang, Chong Chen, Yintao Xu, Lianglun Cheng
Guangdong Provincial Key Laboratory of Cyber-Physical System
Guangdong University of Technology
Guangzhou, China
wangtao_cps@gdut.edu.cn

Abstract—In the narrow industrial welding scene, it is difficult for the 6DOF manipulator to realize intelligent obstacle avoidance planning. This paper proposes an adaptive reinforcement learning on the path planning method of welding manipulator to find a collision-free path in the limited scene. The sub-actor network is designed to conduct guided search on the main actor-network to achieve effective obstacle avoidance. The overestimation of Q value is alleviated by embedding the return distribution function into maximum entropy to replace the shear-double Q learning of SAC. We evaluate our approach on a group path planning experiment, the experiments demonstrate that our method increases learning efficiency and obtains safer policies.

Keywords-Maximum entropy; Welding robot planning; Narrow space planning; Obstacle avoidance

I. INTRODUCTION

The Welding tasks exist in industrial manufacturing processes. With the development of industrial robot technology, automatic welding is gradually replacing manual welding, and a welding manipulator is equipped in the ship production line [1, 2]. However, path planning for specific welding tasks such as spot welding or arc welding, often done by skilled welding technicians through off-line programming instruction, remains a challenge for automated welding. When dealing with the complex welding task of the workpiece with different shapes, the online teaching and offline programming are quite tedious because of the switch of the welding torch's various postures. To satisfy the demand for flexible and high-efficiency welding, a reliable, safe and automatic path planning technology for welding torch position and pose switching is urgently needed.

Collision-free path planning is a well-established subfield of robotics. Compared with mature planning algorithms, such as RRT connect algorithm, the target point selection is too difficult. Under the constraint of narrow space, the six axis robot arm can't find the executable solution at all, so it can't generate a tree. On the contrary, Deep Reinforcement Learning (DRL) can find better solutions through self-learning and trial and error of neural networks. The free space of the right angle welding channel is very small. At least in the one-dimensional range, the distance between obstacles is very small. At the same time, in the industrial welding scene, the weld and obstacles coincide, and the learning difficulty of collision avoidance strategy increases sharply. Therefore, the possibility of a 6DOF robot generating an effective path is very low. In recent years, due to the amazing performance of various robot controllers or planners based on DRL described in [3, 4], many researchers have turned to deep reinforcement learning[5-7]. In order to address the issue mentioned above, this paper aims to solve the path planning problems in high-dimensional continuous states and action spaces.

Compared with mobile robots, welding manipulators with high-dimensional continuous motion space are not easy to be directly processed by deep-RL algorithm because of their huge exploration space and low sampling efficiency. The core of the reinforcement learning algorithm is the tradeoff between exploration and utilization, which limits the convergence of the algorithm. Therefore, the balance between exploration and utilization must be considered when using deep-RL for path planning. In practice, almost all RL algorithms have inaccurate estimates. On the one hand, since the real q value is initially unknown [8, 9], all algorithms introduce some estimation bias and variance. On the other hand, function approximation errors are usually unavoidable. This is particularly problematic because inaccurate estimates can cause any suboptimal behavior to be overestimated, leading to suboptimal strategies.

DRL control of robots to solve complex operational tasks [12, 13] has become an active research area. Specifically, model-free RL has been well studied for robotic manipulative tasks such as grasping and placing objects [14], manual dexterity [15] and peg insertion [16]. To solve long-term tasks, the hierarchical RL approach extends RL algorithms with temporal abstractions, such as options, modular networks and low-level policies for target conditions [17, 18]. Han et al. [19] introduced Deep RL into automatic collision avoidance strategy in three-dimensional space. This algorithm enables the drone to learn the optimal strategy and reach a destination without crashing the drone. By adding the penalty function to the objective function, Mahdavian et al. [20] used a genetic algorithm to optimize the motion of the robot arm, reduce its energy consumption, and realize obstacle avoidance for the robot. Less attention is paid to obstacle avoidance and path planning of manipulators in a complex environment.

These methods are usually used in controlled, spacious environments, and often require extensive examples. In the right-angle welding of the 6DOF manipulator environment, the closer the manipulator is to the target point, the smaller the operating space is, which makes exploration challenging. Before a learning signal is received, this problem is exacerbated in situations where feedback is scarce and considerable exploration is required. Yang et al. [21] used...
Soft Actor Critic (SAC)[22], which is a state-of-the-art method for model-free RL, and random network distillation to capture a target accurately in a simulation. On the basis of the originally expected reward maximization, the maximum entropy framework is introduced, and the participants' goals are expected reward maximization and entropy maximization. In other words, try to be as random as possible when completing tasks. In DRL, the idea of maximum q value and maximum immediate reward is adopted in the selection of actions to ensure the optimization in the overall situation, but it is not considered locally. Especially in the obstacle avoidance planning of 6DOF manipulator at high latitude, the manipulator may directly pass through the edge of the obstacle to obtain the maximum expected reward. However, the mainstream obstacle avoidance strategy combines the RGB-D, which increases the cost, At present, there is no better method for obstacle avoidance strategy without considering visual information.

In order to solve the problem that 6DOF manipulator is difficult to design obstacle avoidance strategy for welding in a narrow space without visual information. This study investigates the application of DRL based on maximum entropy in the industrial welding scene. The main contributions of this paper are as follows: 1) Embedding the action value regression function into the maximum entropy distribution RL, learning the continuous distribution of the state action regression Q value, solving the traditional high Q value estimation RL algorithm and the local exploration and utilization of environmental information by the manipulator. 2) Design the primary and secondary action network structure to improve the effective sampling of the manipulator. 3) Provide a solution for the unique action, state and q-valued strategy is calculated.

The objective function of maximum entropy

\[ J(\pi) = -\min_{\pi} \mathbb{E}_{\pi} \left[ \sum_{t=0}^{\infty} \gamma^t r(s, a, \pi) + \alpha H(\pi(s)) \right] \] (1)

Where, \( \alpha \) is the temperature term, which determines the importance of entropy relative to the reward, thus controlling the randomness of the strategy. Adding entropy has the following benefits: the introduction of entropy encourages exploration while abandoning unpromising paths. Policies can capture multiple near-optimal behavior patterns. Previous work has observed that entropy leads to better exploration. Compared with the traditional objective function, the new objective function can improve the learning speed.

Entropy is used to measure the randomness of random variables[23], and its random distribution is directly considered in the actual calculation. Now to calculate the entropy of \( \pi \), and if \( \pi \) follows a distribution, then the entropy as :

\[ H(\pi(s)) = -\mathbb{E}_{\pi} \log \pi(a|s) \] (2)

The goal is to maximize the cumulative reward of entropy

\[ \pi^* = \arg \max_{\pi} \mathbb{E}_{\pi} \left[ \sum_{t=0}^{\infty} R(s, a, \pi) + \alpha H(\pi(s)) \right] \] (3)

It can be seen that compared with the original RL algorithm, MERL only adds one more entropy term after the reward so that the strategy maximizes the entropy of the strategy while maximizing the cumulative benefit.

3) Soft Policy iteration: In the strategy evaluation stage, the value function corresponding to the strategy is calculated according to the maximum entropy target formula of SAC[23]. For the fixed strategy, its soft Q-value passes the corrected Behrman operation:

\[ Q(s, a) = E[R(s, a) + \gamma \pi(a'|s')] \] (4)

\[ V_{\text{opt}}(s) = E[Q(s, a) - \gamma \log \pi(a|s)] \] (5)

where, \( V(s) \) is soft state value function, then soft policy evaluation can iterate through \( Q_{\text{opt}} = \gamma Q_{\text{opt}} \), and finally, Q converges to a soft Q function of strategy \( \pi \).

\[ \pi_{\text{opt}} = \arg \min_{\pi} D_{KL}(\pi||\pi_{\text{old}}) \exp(Q_{\text{opt}}(s, a)) \] \[ Z_{\text{opt}}(s) \] (6)

Firstly, the q-valued function is transformed into a probability distribution to represent the strategy, and then the strategy with the minimum KL divergence of the strategy and q-valued strategy is calculated. \( \pi_{\text{opt}} \in \pi \), \( \pi_{\text{opt}} \) is a strategy of the optimization formula, which also exists in the strategy space \( \pi \). For all \((s, a) \in S \times A\), \( Q_{\text{opt}}(s, a) > Q_{\text{opt}}(s, a) \), so that each update of the new policy is better than the old policy.

The above soft strategy iterative process is derived based on the tabular environment. For continuous cases, it is necessary to introduce function approximation. Firstly,
the soft state function, soft Q value and strategy function are defined, and corresponding parameters are $\psi$, $\theta$ and $\varphi$ respectively. Then, the objective function of soft state value is

$$J_\psi(\varphi) = \mathbb{E}\left[\frac{1}{2}(V'_\psi(s_t) - E[Q(s_t, a_t) - \alpha \log \pi(a_t | s_t)]^2)\right]$$  \hspace{1cm} (7)$$

The objective function of the soft Q-valued function is

$$J_\theta(\theta) = \mathbb{E}\left[\frac{1}{2}(Q_\theta(s_t, a_t) - Q^\text{target}(s_t, a_t))^2\right]$$ \hspace{1cm} (8)$$

The objective function of policy update is

$$J_\theta(\theta) = \mathbb{E}[Q_\theta(s_t, a_t) - \alpha \log \pi(a_t | s_t)]$$ \hspace{1cm} (9)$$

B. Sub-Actor Network

For right angle welding, the welding must start from the inner corner of the right angle, as shown in the welding gun in front of Fig.1. However, for the robot arm, it can approach the welding point from both sides, as shown in Fig.1 (front and rear). Only one side is a valid sample. Therefore, the primary and secondary areas are designed to avoid invalid sampling of the mechanical arm, so as to improve the sampling efficiency. Therefore, the primary and secondary areas are designed to avoid invalid sampling by the mechanical arm, so as to improve the sampling efficiency. The structure of the sub-actor network is the same as that of the main network, but the difference is that the definition of state space is different. Assuming that the welding point is $(x_0, y_0, z_0)$, the sample space where the welding point is located is divided into two regions, the red primary region $\Omega_{\text{prim}}$ and the blue secondary region $\Omega_{\text{sec}}$ as shown in Fig.2. When the end of the manipulator is in the red region, the action is obtained from the main actor network. When the end of the manipulator is in the blue area, the action is obtained from the sub-actor network, and the sub-actor network can transfer $(s, a, r, s', d)$ to the main network for learning.

$$\begin{cases} \Omega_{\text{prim}} & x > x_0, y > y_0 \\ \Omega_{\text{sec}} & \text{otherwise} \end{cases}$$ \hspace{1cm} (10)$$

The reward function for the sub-network needs to be re-defined, it also uses a nonlinear piecewise function to calculate this part of the reward:

$$r_{\text{sub}} = \begin{cases} -\frac{1}{2}(d_{\text{sub}}^2) & , d_{\text{sub}} \leq \delta \\ -\delta |d_{\text{sub}}| - \frac{1}{2}\delta - d_{\text{sub}}^2 & , d_{\text{sub}} > \delta \end{cases}$$ \hspace{1cm} (11)$$

Where $d_{\text{sub}}$ represents the distance between the mapping of the endpoints of the manipulator in the red area and itself. $\delta$ is the threshold of the function.

Figure 1. Schematic diagram of effective and ineffective areas of welding

The mapping is as follows:

$$\begin{align*} x_o = -x_0, y_o = -y_0 & \quad \text{if } x_0 < px, y_0 < py \\ x_o = x_0, y_o = -y_0 & \quad \text{if } x_0 \geq px, y_0 < py \\ x_o = -x_0, y_o = y_0 & \quad \text{if } x_0 < px, y_0 \geq py \end{align*}$$ \hspace{1cm} (12)$$

C. Return Distribution for Reducing Overestimation

Firstly, to assume the random returns $Z(s,a)$ obey a Gaussian distribution $Z(\cdot | s,a)$. Suppose the mean (i.e., Q-value) and standard deviation of the Gaussian distribution are approximated by $Q_\theta(s,a)$ and $\sigma_\varphi(s,a)$, with parameters $\theta$ and $\psi$, i.e. $Z_{\text{sub}}(\cdot | s,a) = N(Q_\theta(s,a), \sigma_\varphi(s,a)^2)$. Similar to standard Q-learning, a random greedy target $y_{\text{target}} = r + Z(\cdot | s,a)$ is defined, where $a^* = \arg \max_a Q_\theta(s,a')$. Suppose $y_{\text{target}} = \sigma_\varphi(s,a)^2$, which is also assumed to be a Gaussian distribution. Note that even if $Z(s,a)$ and $y_{\text{target}}$ are not strictly Gaussian, we can still use the Gaussian to approximate their distributions, which will not affect the subsequent analysis. Since $\mathbb{E}[y_{\text{target}}]=\mathbb{E}[r]$, and $\mathbb{E}[\sigma_\varphi(s,a)^2]=\mathbb{E}[\sigma_\varphi(s,a)]$, it follows $Z(\cdot | s,a) = N(y_{\text{target}}, \sigma_\varphi(s,a)^2)$. Considering the loss function under the KL divergence measurement, $Q_\theta(s,a)$ and $\sigma_\varphi(s,a)$ are updated by minimizing

$$D_{\text{KL}}(\mu_{\text{approx}}(\cdot | s,a), Z_{\text{sub}}(\cdot | s,a)) = \log \frac{\sigma_\varphi(s,a)}{\sigma_{\text{approx}}} + \frac{(y - Q_\theta(s,a))^2}{2\sigma_\varphi(s,a)^2} - 1$$ \hspace{1cm} (13)$$

$$\theta_{\text{new}} = \theta + \beta \left[\frac{y - Q_\theta(s,a)}{\sigma_\varphi(s,a)}\right] \nabla_{\theta} Q_\theta(s,a)$$ \hspace{1cm} (14)$$

$$\varphi_{\text{new}} = \varphi + \beta \left[\frac{1}{\sigma_\varphi(s,a)^2} \frac{(y - Q_\theta(s,a))^2}{\sigma_\varphi(s,a)}\right] \nabla_{\varphi} \sigma_\varphi(s,a)$$ \hspace{1cm} (15)$$

where $\Delta \sigma = \sigma_{\text{approx}} - \sigma_\varphi(s,a)^2$. Compared with standard Q-learning, $\sigma_\varphi(s,a)$ play a role of adaptively adjusting the update step size of $Q_\theta(s,a)$. In particular, the update step size of $Q_\theta(s,a)$ decreases squarely as $\sigma_\varphi(s,a)$ increases.
D. Environment Modeling

To solve the path planning problem using the deep-RL framework, the original problem must be transformed into a sequential decision problem, namely the Markov decision process (MDP). To do this, it is necessary to define the interface between the agent and the environment.

In practical applications, RL usually chooses some important features as state information. It should be pointed out that the state information of deep-RL represents the environmental information perceived by the agent, implying the influence brought by the agent's behavior. State information is the basis for the agent to make decisions and evaluate its long-term benefits, i.e. cumulative returns. The state design directly determines the convergence, convergence speed and final performance of the deep-RL algorithm.

1) State-space: In this study, the goal of the collision-free path planning task is to train an agent to drive the joints of the welding manipulator without collision, so as to minimize the distance between the end-effector of the welding manipulator and its target position. According to the given task, the state observed from the environment at the time step \( s_t \) is designed as a stacked vector:

\[
s_t = [q, p, d_{\text{ori}}, \text{collision}] \in S
\]

where \( q \in \mathbb{R} \) is the joint position of the welding manipulator, and \( p \in \mathbb{R}^3 \) is the Cartesian coordinate position of the end-effector coordinate system relative to the target coordinate system. \( d_{\text{ori}} \) represents the Euclidean distance from the origin of the end-effector frame to the origin of the target frame. \( \text{collision} \) is a Boolean value, indicating whether each link of the welding manipulator colliders with obstacles.

2) Action space: When applying DRL algorithms to real problems, the most intuitive part is probably the definition of action space \( A \). In this study, the actor in the environment is a 6-DOF welding manipulator. Therefore, the control method of the agent on the manipulator can be intuitively position control, as shown below:

\[
a_t = q_t \in A
\]

where \( q_t \in \mathbb{R}^n \) is the target angular position within the limitation of each rotation joint driven by a servo motor.

The safety constraints of the manipulator control are limited to the operable range of the manipulator body for each joint, that is, the range of joints, as shown in the following Table I.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Joint 1</td>
<td>-160–168</td>
<td>Joint 4</td>
<td>-180–180</td>
</tr>
<tr>
<td>Joint 2</td>
<td>-170–75</td>
<td>Joint 5</td>
<td>-108–108</td>
</tr>
<tr>
<td>Joint 3</td>
<td>40–265</td>
<td>Joint 6</td>
<td>-360–360</td>
</tr>
</tbody>
</table>

3) Reward signal: The reward signal determines whether the agent can finally learn the required skills, which, like the state space, directly affects the convergence speed and final performance of the algorithm. Considering our question, the reward signal can define into three parts.

- The first part is designed for the end-effector to approach a specific target position. The reward uses a nonlinear piecewise function to calculate this part:

\[
r_1 = \begin{cases} n \left( \frac{1}{2} d^2_{\text{ori}} \right) & , \quad d_{\text{ori}} \leq \delta \\ -\delta (d_{\text{ori}} - \frac{1}{2} \delta) & , \quad d_{\text{ori}} > \delta \\ \end{cases}
\]

(18)

Where \( d_{\text{ori}} \) represents the Orientation distance from the origin of the end-effector frame to the origin of the target frame. \( \delta \) is the turning point of the function.

- The second part is designed to avoid collisions, so in this part, we give penalties based on the current collision situation:

\[
r_2 = \begin{cases} -\frac{1}{2} d^2_{\text{coll}} & , \quad d_{\text{coll}} \leq \delta \\ -\delta (|d_{\text{coll}}| - 5\delta) & , \quad d_{\text{coll}} > \delta \\ \end{cases}
\]

(19)

Where \( d_{\text{coll}} \) is the Euclidean distance from the collision point to the target position.

- The third part is relative to the norm of the current action \( q \) shown in Eq.17, which means we expect the agent to take smaller actions for safety at the end.

\[
r_3 = -\|q\|
\]

(20)

Finally, the total reward can be constructed as follows:

\[
r = \lambda_1 r_1 + \lambda_2 r_2 + \lambda_3 r_3
\]

(21)

where \( \lambda \in \mathbb{R} \) denotes the constant scaling factor which is designed to make the contribution of each reward component to the final reward at a close level, so that each target reward can be taken into account by the agent to the same extent.

III. CASE STUDY AND SIMULATION

This paper design our experimental evaluation to answer the following questions: (1) Can SDAC solve complex tasks in obstructed environments more efficiently than conventional RL algorithms? (2) Does SDAC have stronger robustness? (3) Does SDAC learn the optimal path?

A. Environments

In this section, the results obtained by applying the proposed SDAC algorithm will be applied to an industrial manipulator with six joints whose end-effectors must be welded at the right angle weld point, as shown in Fig.4. During its mission, it must avoid collision with the solder plate. For evaluation, the physical environment of the training process was recreated using the Pybullet simulator. The simulator and SDAC learning algorithm are implemented using TensorFlow, using Pybullet remote API
function to interface with Gym in Python. The manipulator used is HSRJR612.

Figure 4. The interaction environment of welding among the steel plate.

B. Baselines

To compare the performance of our method with the following approaches:

- **DDPG**: A policy trained to predict displacements in the robot’s joint angles using DDPG, a deterministic policy RL algorithm.

- **TD3**: A policy trained to predict displacements in the robot’s joint angles using TD3, a twin delayed deep deterministic policy gradient algorithm.

- **SAC**: A policy trained to predict displacements in the robot’s joint angles using SAC, a state-of-the-art model-free RL algorithm.

- **SDAC (Ours)**: Our method predicts a joint displacement in the extended action space.

In all experiments, the total reward and average loss functions for each episode were tracked. In addition, in each step of the simulation, the distance norm from the end to the target and the distance norm from the obstacle are obtained to better demonstrate the behavior of the robot in a training set. All training sessions are conducted on a machine equipped with 8 * Intel (R) Xeon(R) CPU E5-1620 V3 @ 3.50GHz, 32GB RAM and NVIDIA Quadro K5000 GPU, 4GB DRAM. Each process takes about four hours. As you can see from Fig.5, the cumulative reward converges to a maximum in all the experiments: this means that the robot has learned an efficient way to complete the task. This also applies to trends in the average loss function, which show variations.

Table II. Choice of Parameters for the Experiments

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>critic_lr</td>
<td>8e-5</td>
<td>delay_update</td>
<td>2</td>
</tr>
<tr>
<td>actor_lr</td>
<td>5e-5</td>
<td>α</td>
<td>5e-5</td>
</tr>
<tr>
<td>end_lr</td>
<td>1e-6</td>
<td>r₃</td>
<td>100</td>
</tr>
<tr>
<td>γ</td>
<td>0.99</td>
<td>r₃</td>
<td>0.1</td>
</tr>
</tbody>
</table>

B. Results

First, the goal of the deep-RL algorithm is to maximize the cumulative reward time step in a finite way within an episode. Therefore, it is necessary to analyze the trend reward curve or the comparison number of learning curve and training, which reflects whether the goal-deterministic strategy model converges and the learning efficiency. As shown in Fig.5, DDPG, SAC and TD3 begin to converge after 400 episodes, but they are prone to reward mutation in the later training process. It is because the end of the manipulator is close to the target point and close to the welding plate in the later training period, and the collision-free space suddenly shrinks, resulting in a large increase in the probability of collision. It is difficult to find a safe and stable attitude near the weld. On the contrary, SDAC consists of two actor networks, the primary actor network guides the manipulator to the target point, and the sub-actor network guides the obstacle avoidance, which can adapt to the situation that the collision space becomes suddenly larger. The final results show that DDPG, TD3 and SAC can keep the manipulator at a distance of 2~3cm from the welding point, and cannot find a collision-free path. Our method can stably approach the solder joint, keep the solder joint within 1cm, and find a collision-free path. Fig.6 shows the planning sequence of this method. The mechanical arm can quickly approach the welding target point, and with the narrowing of free space, the end of the mechanical arm can find a non-collision path to reach the target point in the narrow space precisely.

Figure 6. The path planning solutions conducted by our method are rendered in Pybullet. Select some pivotal way points to express the complete safe path.

Figure 7. Distance curve of end close to solder joint.
To test the robustness of the algorithm, joint noise is applied to the trained network model. The joint noise is applied in a step of each episode. In the presence of joint noise, whether the end of the manipulator can approach the target point safely and stably is regarded as the evaluation criterion. Fig.7 (left) shows the distance curve from the end to the solder joint (collision-free path) tested more than 1000 times, and Fig.7 (right) shows the distance curve after adding noise. The results are obtained through the trained network model. The results show that the success rate of our robot arm's collision-free planning to reach solder joints is more than 90%.

The method proposed in this paper is applicable to the speed planning of the six-axis manipulator, and can plan a collision-free path in a narrow environment. Compared with the mature sampling algorithm, it avoids the complex inverse kinematics calculation and ensures robustness. However, there are still some shortcomings in this paper. In the actual process, the constraints of the manipulator's own motor, such as torque and acceleration, are not considered. In addition, the welding environment needs to be accurately modeled, otherwise, there may be errors in path planning.

IV. Conclusion

In this paper, a collision-free path planning algorithm for right-angle welds of 6DOF manipulator based on SAC is successfully established. The results show that this method not only improves the convergence performance of the algorithm, but also has good robustness. However, in the case of blindness, the welding track overlaps with the obstacles, and the depth and other information cannot be obtained. The collision avoidance strategy designed in this paper provides an effective solution. In addition, trajectory planning does not take into account the dynamic constraints of the manipulator itself. The potential direction of future work includes multi-task planning so that it can be tested in more complex scenes, not just in a single point-to-point environment. At the same time, dynamic constraints are added to make the manipulator movement more in line with the actual scene.

Acknowledgment

Our work is supported by multiple funds in China, including the Key Program of NSFC-Guangdong Joint Funds (U1801263), Industrial core and key technology plan of ZhuHai City (ZH22044702190034HJL), and our work is also supported by Guangdong Provincial Key Laboratory of Cyber-Physical System (2020B1212060069).

References


Safe, efficient waypoint manipulation for path planning of non-holonomic robots

1st Akhilesh Bhat
NEC Japan
akhibhat09@nec.com

2nd Natsuki Kai
NEC Japan
n-kai@nec.com

3rd Takayuki Suzuki
NEC Japan
takayuki-suzuki@nec.com

4th Takahiro Shiroshima
NEC Japan
t-shiroshima@nec.com

5th Hiroshi Yoshida
NEC Japan
yoshida@nec.com

Abstract—During the pandemic, health and safety concerns kept people indoors and increased the demand for same-day delivery. This put a huge stress on the e-commerce industry. Adoption of warehouse automation skyrocketed, leading to an increased emphasis on efficiency. In warehouses where large industrial robots like forklifts are deployed, efficiency and safety, both are important. Conventional methods like Reeds-Shepp (RS) that consider the motion model are often sub-optimal as they do not consider obstacles, while those that are optimal like Dijkstra and A* are difficult to implement due to kinematic constraints. This paper proposes a hierarchical structure consisting of an A*-based global planner which goes through a pre-processing step. The local planner then generates a Reeds-Shepp curve to a waypoint chosen using a "cost" parameter.

Index Terms—autonomous forklift, collision avoidance, motion planning, non-holonomic robot, safe and efficient

I. INTRODUCTION

Due to the Coronavirus pandemic, online sales accounted for 18% of all retail sales worldwide in 2020 and this figure is projected to reach 21.8% in 2024. This is a huge challenge, in terms of efficiency, for businesses all over the world. COVID-19 caused severe disruptions to supply chain and logistics, affecting the ability of e-commerce businesses to meet customer demands. Health and safety issues have caused warehouses, distribution and fulfillment centers to operate with reduced number of workers, exacerbating the already prevalent labor shortage. A survey done in 2018 by Persol Research and Consulting and Chuo University [1] has predicted that the demand for workers in Japan will reach 70.73 million in 2030 resulting in a shortage of about 6.44 million. In order to compensate for this, companies are adopting robots and other improved technology to increase the overall productivity, accuracy and efficiency in their warehouses. It has been estimated that mobile robotics in material handling and logistics will become a USD 75 billion market by 2027 [2].

Although warehouse automation was adopted “in order to operate with fewer floor workers to keep people safe and socially distant”, a report by the Center for Investigative Reporting revealed, that between 2016 and 2019, Amazon employees in automated warehouse sustained serious injuries at a rate 50% than that at non-automated ones [3].

The current existing methods try to deal with some of the above issues but often fail to tackle all of them at once. The techniques that focus on the safety compensate for the efficiency by generating longer paths. A method that is efficient and safe and can generate a path real-time will help towards making industrial warehouses completely autonomous. We contribute towards this goal by proposing a real-time hierarchical planner that implements a Reeds-Shepp curves based local planner on a pre-processed global plan generated by the default global planner in Robot Operating System (ROS).

The primary contributions of this work are as follows:

• We propose a waypoint cost equation to facilitate the selection of an optimal waypoint to track for a non-holonomic robot with a fixed minimum turning radius.
• Results of the local planner being tested on a Gazebo-based simulator are shown for test cases including variations of start and goal orientations.
• We deploy this planner on a real forklift as shown in Fig.1 and conduct experiments to show the feasibility of real-time implementation.

The remainder of the paper is organised as follows. Section II reviews related literature on various planning algorithms and their drawbacks. Section III presents the individual components of the proposed algorithm. The evaluation and results of experiments done in simulation and on a real forklift are explained in Section IV and V respectively. Section VI concludes the paper.

II. LITERATURE REVIEW

A review of motion planning techniques by Gonzalez Bautista et al. [4] concludes that graph based search algorithms like A* and D* are most often used in real-world situations because of their efficiency, optimality and adaptability. Although experimental and simulation results in Setiawan et al. [5] show...
that D* Lite can plan a shorter path in faster time than A*, Kim et al. [6] show that the D* Lite algorithm is less effective than the A* in relatively smaller and less complex environments. Even so, the vanilla algorithms do not consider the kinematic constraints of the robot on which they are being implemented. This is especially important in material handling applications in warehouses where the robots carry heavy payloads with a risk of tipping over while making turns. Vehicle characteristics like the minimum turning radius are considered in Yang and Wushan [7] where a grid based path-smoothing method is applied to the path generated by A*. However, according to Guruji et al. [8], the computational time of the algorithm tends to increase exponentially with the size of the environment.

Genetic algorithms and Deep learning methods are also slowly being used for path planning but due to a large computational overhead, they cannot be used in environments where safety and quick response is of utmost importance.

Sampling-based planners (SBP) like rapidly exploring random trees (RRT) and probabilistic roadmaps (PRM) have been implemented as they generate fast and feasible solutions in dense environments. [9] surveys state of the art sampling based planners and conclude that the suboptimal paths generated are a major drawback. In order to use the speed of these planners, work has been done to improve the path quality instead [10]. Smoothing techniques using polynomials, Bezier curves, B-splines and Clothoids have been implemented to offset the suboptimality but time spent on such post processing methods makes them ineffective in real-time applications. Karaman and Frazzoli [11] put forth a family of sampling based planners, RRT* and PRM*, which guaranteed asymptotic optimality. However, the convergence rate has been shown to be slow. Luna et al. [12] show that in fact, a post processing method can achieve better results than RRT*. To overcome the slow convergence rate, Anytime RRT* was proposed in [13]. The algorithm was implemented on a robotic forklift and finds a suboptimal path and converges towards optimality within a given planning time. However, this method does not give consideration to the initial and final pose of the non-holonomic robot. Sobue et. al [14] focus on this issue by compensating for the optimality of the path.

Pioneering research [15] [16] has shown that RS path is indeed the shortest curve that connects two points in the 2D Euclidean plane for a car-like robot that can travel both forward and backward. Nevertheless, the generation of RS paths assumes an obstacle-free plane. As a result, these curves cannot be directly used in real-world situations. In order to incorporate obstacle avoidance, RS paths have been modified using continuous curvature (CC) paths [17] and hybrid curvature (HC) steers [18]. Paths generated by the aforementioned methods have been shown to be of sufficient length and also more time consuming. Even though these modified curves can be used as steer functions in SBP, the time taken to reach an optimal solution renders these methods infeasible.

Most algorithms in the literature that focus on improving the safety do so by compensating for the total distance travelled. Offline planners that are capable of generating the shortest path while being safe are not able to do it in real-time, making them infeasible for our application.

III. METHODOLOGY

For our planner, we make certain assumptions:

- The environment of the robot is static. The position of obstacles can change in different runs, but the obstacles are not considered to be dynamic.
- We focus mainly on non-holonomic robots that can move both "FORWARD" and "REVERSE" similar to the Reeds-Shepp Car [16]

The planner is based on a hierarchical structure consisting of a global and a local planner. We use the move_base package in ROS for ease of implementation. The parameters, especially "inflation_radius" and "cost_scaling_factor", for the global costmap and the local costmap are modified to ensure safety of the forklift while navigation. The global planner generates a path consisting of waypoints from the initial pose to a given goal pose. The local planner chooses one of the waypoints in the global planner and moves towards it while avoiding nearby obstacles. The efficiency of the robot’s motion, thus depends a lot on the waypoint chosen by the local planner.

Holonomic and omni-directional robots can choose any point without compensating for the efficiency, but for non-holonomic robots with a non-zero turning radius, the choice of a local goal is important. Due to kinematic constraints of such robots, a point chosen very close to the robot may require unnecessary direction changes and pose adjustments while if chosen further away, it is not entirely safe. Also, due to the presence of long, narrow corridors as well as large empty spaces in warehouse environments, it is important to decide where the robot can take a turn if needed. Defining appropriate orientations to the waypoints in the global plan thus becomes an important pre-processing step.

A. Orientation Assignment

In order to assign orientations to the waypoints of the global plan, it is important to consider the heading angle ($\theta_s$) and the goal approach angle ($\theta_g$) for a non-holonomic robot. These angles, as illustrated in Fig.2 are respectively the angles that the start pose and goal pose make with the global path generated. The similarity or dissimilarity in the specified angles helps decide whether the robot will need to perform a direction change (DC) maneuver or not. When the heading angle and goal approach angle are both $\approx 0^\circ$, the robot can continue moving in the same direction without the need for a DC maneuver. On the other hand, when the heading angle is $\approx 0^\circ$ and the goal angle is $\approx 180^\circ$ the robot needs to perform a DC maneuver before approaching the goal point. A special case occurs when either angle is $\approx 90^\circ$. In this case, a DC maneuver might be required at the respective location.

We thus divide the orientation assignment problem into 3 different cases depending on the goal approach angle.

1) $0^\circ < \theta_g \leq 90^\circ - \theta_m$
2) $90^\circ - \theta_m < \theta_g \leq 90^\circ + \theta_m$
3) $90^\circ + \theta_m < \theta_g \leq 180^\circ$
Here, \( \theta_m \) was introduced to handle the edge cases when the goal angle was near 90\(^\circ\) and its value was determined by observing real-time experiments, considering the size and minimum turning radius of the forklift.

The orientation assigned also depends on whether there is an existing obstacle near the start or goal point which may make a DC maneuver near the point difficult. As shown in Fig.3, we thus define the following 2 orientation modes for the global plan:

- Towards Goal (TG): In this case, the orientation for all the waypoints is set along the global path, pointing towards the global goal.
- Away from Goal (AFG): In this case, the orientation for all the waypoints is set along the global path, pointing away from the global goal.

When no obstacle is present within a circle of pre-defined radius (\( R_{obs} \)) near the start or goal point, the orientation for the waypoints is assigned as follows:

1) \( 0^\circ < \theta_s \leq 75^\circ \) TG
2) \( 75^\circ < \theta_s \leq 105^\circ \) TG/AFG
3) \( 105^\circ < \theta_s \leq 180^\circ \) AFG

In the case when a DC maneuver is required, it is performed starting at the initial pose. In cases when this is not possible due to the presence of an obstacle, we further divide the above cases based on the heading angle value.

1) \( 0^\circ < \theta_s \leq 75^\circ \) TG
2) \( 75^\circ < \theta_s \leq 105^\circ \) TG
3) \( 105^\circ < \theta_s \leq 180^\circ \) AFG

When \( 75^\circ < \theta_s \leq 105^\circ \), the orientation for the waypoints in the global plan is decided on the basis of the heading angle \( \theta_s \). Also, since the planning process takes place at a frequency of 10Hz, the orientation of the waypoints in the global plan switches as soon as the obstacle is out of a circle of \( R_{obs} \).

### B. Waypoint Cost Calculation

To select an optimal waypoint, we estimate the ease of reaching a goal pose by calculating a cost required to reach the waypoint from the initial position. This depends on various factors which are explained as follows:

1) **Euclidean distance (d):** For a robot with a non-zero minimum turning radius, a waypoint needs to be at a distance more than twice its minimum turning radius to be able to reach without having to change the direction of motion more than once. The proximity of the waypoint to the initial position is thus incorporated into its cost by calculating the euclidean distance of the waypoint from the initial position.

2) **Number of Direction Changes (\( N_{DC} \)):** Since the main focus of our algorithm is non-holonomic robots, direction changes need to be factored into the waypoint cost. In this case, we refer to the change from "FORWARD" to "REVERSE" motion as a direction change. Higher the number of times the robot needs to change its direction, more is the time required and higher is the path length, thus reducing efficiency. \( N_{DC} \) is calculated by measuring the number of cusps in the Reeds-Shepp path generated till the waypoint.

3) **Local path length ratio (L):** As it has been clearly established that RS paths do not consider the environmental obstacles, it is necessary to make sure that the local path generated avoids any collisions. We do this by comparing how close the local path is to the corresponding section of the global path. Using the RS path function, we calculate the local path length to the selected waypoint (\( l_{RS} \)) and calculate the ratio of this length to the global path length up to this waypoint (\( l_G \)). Nearer the ratio is to 1, the closer is the local path to the global path. Also, since the global planner being used in this implementation is based on the A* algorithm, a local path similar to the global path guarantees obstacle avoidance.

\[
L = \frac{l_{RS}}{l_G}
\]  

4) **Heading Angle (\( \theta_1 \)):** The heading angle of the robot is calculated as the angle between its current pose and the line connecting its position and the waypoint. A heading angle of 0\(^\circ\) implies a straight path with no DC while increasing the heading angle up to 90\(^\circ\) implies at least one DC. As the heading angle increases further up to 180\(^\circ\), the probability of a DC decreases again as the robot can move in reverse.

5) **Orientation Difference (\( \theta_{diff} \)):** As shown in Fig.4 orientation difference is the absolute value of difference in pose angle of the initial position (\( \theta_1 \)) and the selected waypoint (\( \theta_2 \)), calculated in reference to a common axis. Navigation is easy when orientation difference is low, whereas when orientation difference is high, multiple DC maybe involved increasing path length as well as time required.

6) **Waypoint Cost:** The above factors are combined into a formula as in Eq.2 to calculate the cost for a given waypoint. Hyperbolic tan is used to normalize the values between 0 and 1.
1 so that it is easier to define a threshold for selecting the most suitable waypoint.

\[
\Sigma = \tanh(\alpha d^2 + \beta N_{DC} + \gamma L + \delta \sin(|\theta_s|) + \xi \sin(|\theta_{\text{diff}}|)) \tag{2}
\]

The coefficients for the above equation were determined by varying the values while performing experiments. These values vary according to various factors like minimum turning radius of the robot and environment of operation of the robot.

C. Waypoint Selection

The waypoint needs to be at a suitable distance from the initial position for the robot to be able to reach it without unnecessary DC maneuvers. We thus define a threshold cost \((\Sigma_{th})\) for selecting the goal for the local planner. Starting from the initial pose, we calculate the \(\Sigma\) for each point on the global planner till we find a waypoint with cost lower than the threshold. We choose it as the goal and generate a RS path from the initial pose. While selecting the waypoint, if a direct Reeds-Shepp path exists such that there are no obstacles within a certain radius of each point, then this path is sent to the controller to track. The RS path program that generates the shortest path, is modified to return the next best path available if the shortest path cannot be traversed due to a possibility of collision. The entire process has been summarized in a block diagram as shown in Fig. 5.

IV. SIMULATION EVALUATION

A. Setup

A ROS-based Gazebo simulator was developed for ease of testing the planning algorithm. Two RP-LIDARs were attached on the left and right leg of the forklift respectively. A single Realsense D-435 camera was attached in the front. The in-built parking lot environment in Gazebo was populated with vehicles and other obstacles as shown in Fig.6 for testing the proposed planner.

B. Evaluation

On the simulator, we compared the performance of our planner against the default planner in ROS consisting of an A* based global planner and the TEB local planner [19]. We formulated 5 different start and goal orientations for the 9 different cases as explained in Sec.III-A. These 45 trials were conducted for each planner on the parking lot environment in Gazebo. We recorded the travelled path length and compared it to the global path to get a measure of how close the path travelled by a non-holonomic robot is to an ideal path for a holonomic robot.

Fig. 7 shows 3 instances of trials that were performed on the simulator. The bold red trajectory shows the actual path taken by the forklift when the proposed planner was used, while the dotted green trajectory was generated by the default ROS planner. The trajectory has been plotted against the occupancy map of the simulation environment. In Fig.7(a), when the heading angle and the goal approach angle are both less than 75\(^\circ\), both the planners fare similar but the length is comparatively smaller due to RS curves being used. In Fig.7(b), the heading angle is less than 75\(^\circ\) but the goal approach angle has been changed to greater than 105\(^\circ\). In this case, the default planner starts out in the forward direction till it approaches the goal. It then performs a DC maneuver to align the robot’s position with the goal. On the other hand, the proposed planner is able to evaluate that a DC maneuver will be required and performs it at the beginning. The robot then moves in reverse till it reaches the goal point. Even though both paths amount to approximately the same length, we believe that moving in reverse after performing the DC maneuver is better in large environments when the local map generated is a very small part of the entire map. Also, if the environmental obstacles have changed position, the robot may or may not be able to perform a DC maneuver near the goal point. In the case when both heading angle and goal approach angle are greater than 105\(^\circ\) as shown in Fig.7(c), the performance of the default planner is clearly below par. The robot starts out by performing an unnecessary DC maneuver near the initial position and moves forward till the goal point where another unnecessary DC maneuver is performed. The proposed planner in this case, using the heading angle and
the goal approach angle, evaluates that no DC maneuver is required and commands the robot to move in reverse till the goal point.

Fig. 8 corroborates the obstacle avoidance capabilities of the proposed method. In Fig. 8(a), when there are no obstacles nearby, the local planner generates a path with no DC maneuvers. In Fig. 8(b), however, the newly introduced obstacle blocks the previously generated path. Here, the global planner modifies the global path to avoid the new obstacle. The local planner also modifies the local path introducing a DC maneuver to first move in "REVERSE" and then "FORWARD" to avoid the obstacle.

C. Results

The proposed planner outperforms the default planner especially in the cases when DC maneuvers need to be performed. The preprocessing step helps modify the orientations for the waypoints on the global path thus helping the planner perform much better. Overall, the length travelled by the robot when implementing the proposed planner was found to be 15% shorter than when implementing the default planner. Table I shows the comparison of the path lengths generated by the default ROS planner and the proposed planner for the different start and goal orientations as explained in Section III-A.

The time required to generate a plan was found to depend on the proximity of the robot to obstacles. When there were obstacles near the generated global path within a certain radius, the average time required for planning was 0.3 seconds. On the other hand, when there were no obstacles, the time required reduced drastically to 3.2 milliseconds. It was also noticed during experiments that the default planner first generates a feasible path and then improves upon it iteratively to give the best possible path. As a result, there was a time delay of approximately 5 seconds before the default planner would settle on the best local plan.

V. REAL-WORLD VERIFICATION

A. Hardware Setup

For real-world experiments, we used a reach-type forklift by Nichiyu. The code runs on an AMD Ryzen 7 3800X processor having 8 cores, 4.2 GHz clock speed and 32 GB RAM. An Intel Realsense D455 cameras is installed on the forklift for localization. Also, two RP-LiDAR A3 are installed, one on the left and one on the right, for detecting nearby obstacles and updating the costmap. The experiments were conducted in a parking lot to simulate a structured warehouse environment.

B. Evaluation

In order to validate the real-time working of our planning algorithm, we also performed autonomous missions in a parking lot. An occupancy map of the environment was made beforehand to be used for localization. While ensuring safety, 9 different trials were conducted in a parking lot. These trials included 5 of the 9 cases explained in Section III-A. The start and goal orientations were similar to the experiments performed on the simulator. Odometry data of the forklift was recorded and then compared to the global plan generated. On conducting experiments, it was observed that the planner was able to generate an obstacle-free path in real-time for the forklift. When $\theta_s$ and $\theta_g$ were both less than $75^\circ$, or both more than $105^\circ$, no unnecessary DC maneuvers were generated. In the case when $75^\circ < \theta_s \leq 105^\circ$, it was observed that an extra DC maneuver was generated at the initial pose. This was attributed to a higher minimum turning radius of the robot. We plan to implement a waypoint-shift technique to reduce this extra DC maneuver as future work.

VI. CONCLUSION

In this work, we have proposed a new algorithm for implementation on large industrial robots like forklifts operating in autonomous warehouses. We have demonstrated...
through experiments, both in simulation as well as on a real forklift, that large robots can move safely in warehouse-like environments without having to compensate on the efficiency of motion. Comparing our algorithm with the default algorithm in ROS, we have shown that the path generated by the proposed planner is on average 15% shorter. The application of this algorithm can be further extended to the motion of large robots in construction sites which are often unsafe for humans. We also believe that this algorithm can be applied to a more general class of robots with a non-zero minimum turning radius. Planning in the vicinity of dynamic obstacles has been left for future work.

REFERENCES


<table>
<thead>
<tr>
<th>$\theta_g$</th>
<th>$\theta_s$</th>
<th>$l_{global}$ [m]</th>
<th>$d_{def}$ [m]</th>
<th>$d_{pro}$ [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\theta_g \leq 75^\circ$</td>
<td>$\theta_s \leq 75^\circ$</td>
<td>14.24</td>
<td>16.06</td>
<td>14.46</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14.36</td>
<td>17.16</td>
<td>15.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14.55</td>
<td>17.21</td>
<td>15.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.61</td>
<td>13.4</td>
<td>12.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.84</td>
<td>17.25</td>
<td>13.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.54</td>
<td>16.02</td>
<td>14.49</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.75</td>
<td>17.37</td>
<td>14.92</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.91</td>
<td>16.78</td>
<td>15.05</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.18</td>
<td>16.47</td>
<td>14.29</td>
</tr>
<tr>
<td>$75^\circ &lt; \theta_g \leq 105^\circ$</td>
<td>$\theta_s \leq 75^\circ$</td>
<td>10.95</td>
<td>14.66</td>
<td>12.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.06</td>
<td>17.44</td>
<td>15.71</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.21</td>
<td>23.0</td>
<td>19.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.18</td>
<td>17.66</td>
<td>13.11</td>
</tr>
<tr>
<td></td>
<td></td>
<td>7.69</td>
<td>14.49</td>
<td>8.39</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11.62</td>
<td>17.09</td>
<td>15.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.44</td>
<td>17.59</td>
<td>13.54</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.36</td>
<td>18.53</td>
<td>12.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.17</td>
<td>13.7</td>
<td>9.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.01</td>
<td>21.34</td>
<td>19.89</td>
</tr>
<tr>
<td>$\theta_g &gt; 105^\circ$</td>
<td>$\theta_s \leq 75^\circ$</td>
<td>14.49</td>
<td>20.14</td>
<td>18.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>12.06</td>
<td>18.63</td>
<td>15.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td>13.73</td>
<td>18.35</td>
<td>14.64</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10.20</td>
<td>17.29</td>
<td>11.60</td>
</tr>
<tr>
<td></td>
<td></td>
<td>9.47</td>
<td>14.46</td>
<td>10.81</td>
</tr>
<tr>
<td></td>
<td></td>
<td>8.20</td>
<td>11.79</td>
<td>9.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>14.26</td>
<td>20.14</td>
<td>14.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15.57</td>
<td>19.51</td>
<td>13.43</td>
</tr>
</tbody>
</table>

TABLE I: Simulator trial results
Multi-Task Learning for Time Series Forecasting Using NARMAX-LSTM

Nerfita Nikentari, Hua-Liang Wei

Department of Automatic Control and Systems Engineering
University of Sheffield, Sheffield

nnikentari1@sheffield.ac.uk, w.hualiang@sheffield.ac.uk (*corresponding author: H.-L. Wei)

Abstract—Traditional time series forecasting models based on neural networks are usually designed for a single task, which ignores the source of knowledge or information from other related tasks. On the other hand, multitask learning (MTL) offers a new perspective where one model is not only for one forecasting case or task but several tasks at the same time. The MTL model proposed in this study is based on the Nonlinear AutoRegressive Moving Average with eXogenous input (NARMAX) model and Long-Short Term Memory (LSTM) Neural Networks. The NARMAX-LSTM model is tested and validated using three different time horizons and different cases/tasks. The model forecasting performance is compared with that of three other linear and nonlinear models. The comparison results show that the multi-task NARMAX-LSTM model does not only have better prediction performance but also avoid suffering from negative transfer and overfitting.

Keywords—multitask learning, NARMAX, LSTM, forecasting;

I. INTRODUCTION

Machine learning provides an important tool for solving real-life data based modelling problems. Traditionally, different modelling tasks are usually solved separately by choosing a specific technique for the task to find optimal solutions. Such a modelling approach is called single task learning (STL). The STL method can only give one forecast result using an available model; it requires to fine-tune other models in cases where there is a need to produce other forecast results for a similar or related task.

However, in certain cases, it is possible to simultaneously deal with several tasks/problems by exploiting the information and knowledge from other similar and related tasks [1, 2]. Such an approach is called Multi-Task Learning (MTL). The term Multi Task is used to describe a single model that can be used to conduct two or more tasks at the same times [3]. The MTL framework attempt to share useful information across tasks in parallel. The goal of MTL is to leverage information in various learning tasks to improve the model generalisation performance [4, 5]. From this perspective of performance generalisation, the high computational capacity of deep neural networks (DNN) can be combined with MTL to improve its performance. This approach has been widely applied in numerous area of science and engineering [2,5].

One of the benefits of neural networks (NN) is their capability to perform various data sets and endpoints concurrently. Caruana [6] argued that neural networks are suitable for MTL problems because NN can learn hidden information of the data shared between multiple tasks [3, 7]. The basic idea of combining NN with MTL is as follows: the NN hidden layer are shared by all tasks (targets/outputs); the training or learning process for all task is implemented in parallel; some hidden nodes may be specially designed for a specific task; add auxiliary output nodes for auxiliary tasks [8, 9].

NARMAX is a class of non-linear model that can be treated as a neural network. This study uses LSTM neural network as non-linear framework for NARMAX [10].

The combination of MTL can increase the effectiveness of the resulting models by augmenting the training set and by using different datasets for related tasks. Because the model has to learn shared information for multiple tasks, there is a lower possibility of overfitting [10]. Another advantage of MTL is that it will cut down the number of networks that need to be trained, optimised, and evaluated. Therefore, MTL would be more cost-effective in terms of computational resources. In the industrial world, this plays an important role in cost-saving when dealing with DNN [3]. Although MTL has the aforementioned advantages, it has drawbacks and one of them is the negative transfer [11]. The model performance may be better on average overall tasks using the MTL setting, but for some particular tasks, MTL may produce unsatisfactory performance compare to independently trained framework or STL. Such a detrimental effect in MTL performance is called negative transfer. Negative transfer can also be described as the situation where accurate forecasting for less difficult tasks are negatively affected by inaccurate forecasting results for harder tasks [2, 12, 13].

This study proposes a novel MTL framework based on NARMAX-LSTM that can simultaneously forecast several tasks to reduce the complex process of designing the neural network models and improving the forecast accuracy using models that avoid or overcome experience negative transfer and overfitting.

399
The main contributions of the work are summarized as follows:

1) The proposal of a novel MTL framework using NARMAX-LSTM that addresses the challenges of overfitting and negative transfer problems.

2) The design and scheme of a single model structure that can be used for multiple tasks/cases (e.g., forecasting) simultaneously.

3) The analysis of the performance of MTL NARMAX – LSTM, and the comparison with single-task baselines for three different time horizons.

II. THEORETICAL BACKGROUND FOR MULTITASK LEARNING AND NARMAX

A. Multitask Learning

MTL is a machine learning approach for simultaneously solving and learning multiple tasks that related to each other by shared knowledge among the tasks [5, 14]. MTL is also called learning with auxiliary tasks, jointly learns and learns to learn [15]. The auxiliary tasks are expected to help the main task train or learn its tasks by supplying knowledge and representation [16]. The definition of MTL is as follows:

Given $m$ learning tasks $\{T_i\}_{i=1}^m$, where the subset of the tasks or all the tasks are related but not exactly alike, multitask learning aims to improve the learning of a model for $T_i$ by using the information contained in the $m$ tasks.

MTL can be seen as a machine learning way to mimic the activity of how humans learns. The information or knowledge that we acquire from somewhere can often be transferred to and/or used somewhere else. For instance, skills learned from a sport activity can usually be applied to another sport, such as skills learned in squash and tennis can benefit each other [4, 15, 16]. MTL is useful for dealing with a data modelling problem under the following scenario: a set of tasks have commonalities between each other; therefore, the model for each task could gain advantages by sharing knowledge across the tasks than training then independently [5, 16].

MTL methods can be roughly categorized into two classes soft parameter sharing and hard parameter sharing [15, 16, 17]. The fundamental idea of hard parameter sharing is sharing the hidden layer among all tasks and having output separately or allocated several specific output layers for specific tasks [15, 17, 18]. This approach dramatically impacts the risk of overfitting and negative transfer problems. These two approaches are not ideal enough to accurately describe the multitask expertise [7, 15]. The hard parameter sharing architecture is shown in Figure 1.

![Figure 1. Hard parameter sharing architecture](source: adapted from [15, 18])

B. NARMAX

NARMAX is transparent non-linear system identification method which focuses on determining the best model structure including selecting most useful and important model terms based on available systems output and output data [10]. The NARMAX model is expressed as follows:

$$y(t) = f(y(t-1), y(t-1), ..., y(t-n_{y}), u(t-d), u(t-d-1) ..., u(t-d-n_{u}), e(t-1), e(t-2) ..., e(t-d-n_{e}) + e(t)$$

(1)

where $y(t)$, $u(t)$ and $e(t)$ are the system output, input, and noise sequences, respectively; $n_{y}, n_{u}, n_{e}$ are the maximum delays of the output, input and noise; $f(\cdot)$ is nonlinear function or mapping, and $d$ is a time delay typically set to $d=1$. The problems or tasks that is relevant to nonlinear systems identification are also suitable to neural networks.

The nonlinear mapping used in this study is based on LSTMs. LSTMs as nonlinear neural networks, which can be used to deal with nonlinear time series modeling tasks.

III. THE PROPOSED MODEL

A. Model

The proposed NARMAX-LSTM model is designed based on the characteristics of the problem to be solved. If the input data pattern changes over time, like in a time series modelling problem, the feed-forward neural network may not perform well. This underperformance results from that the preceding output data are not being
used as input to derive the relationship between the input and output.

This study uses a direct approach to generate forecast for time series data. The direct approach uses a separate model for each fixed horizon h of interest [21,22,23]. The h steps prediction of tide level forecasting using NARMAX-LSTM as seen in [21] can be define as:

\[ y(t) = f \left( y(t-h), y(t-h-1), ..., y(t-n_x), u(t-1), u(t-2), ..., u(t-n_u), e(t), e(t-1), e(t-2), ..., e(t-n_e) \right) + e(t) \]  

(2)

The multi-task model has NARMAX-LSTM for direct forecasting approach can be represented as:

\[ y_i(t) = f_i \left( y_1(t-h), y_2(t-h-1), y_3(t-h-n_x), ..., y_m(t-h-n_x), y_m(t-h-n_x), u_1(t-1), u_2(t-2), ..., u_m(t-n_u), e_1(t-1), e_2(t-2), ..., e_m(t-n_e) \right), i = 1, ..., m \]  

(3)

Note that in this study the NARMAX model is realized based on LSTM, the functions \( f_i \) \( (i = 1, 2, ..., m) \) are derived from the mathematical model of LSTM. LSTM for NARMAX model can be expressed as:

**Forget Gate:**  
\[ f(t) = \sigma \left( W_f [y(t-d), u(t-d), e(t-d)] + b_f \right) \]  

(4)

**Input Gate:**  
\[ i(t) = \sigma \left( W_i [y(t-d), u(t-d), e(t-d)] + b_i \right) \]  

(5)

**Input Gate:**  
\[ c(t) = f(t), c(t-1) + i(t), \hat{c}(t) \]  

(6)

**Output Gate:**  
\[ o(t) = \sigma \left( W_o [y(t-d), u(t-d), e(t-d)] + b_o \right) \]  

(7)

where \( \sigma \) is the sigmoid function; \( \tanh \) is the tangent function; \( i \) is the input gate; \( f \) is the forget gate; \( o \) is the output gate; \( c \) is the memory cell; \( h \) is the hidden layer output; \( x_t, h_t \) is the input vector and hidden vector at time step \( t; W \) denotes the weight matrix and \( b \) denotes the bias.

The sharing layer is located in the hidden layer. The remaining layers are divided into different specific tasks, as illustrated in Fig. 1. The LSTM layer acts as hidden layer and is shared all their properties with all four tasks. The lower layer is task-specific layer which is split according to their specific task. Note that in this study multi-task NARMAX-LSTM model is designed for tide level forecasting, more details are given in the next section.

IV. EXPERIMENTS

A. Data Preparation

This study is concerned with tide level prediction. The experimental dataset used consist of about 29,000 samples. The data are time series of tide level recorded every 15 minutes. This study considered three different forecast horizons: one step, two steps ahead and three steps, corresponding to 15 minutes, 30 minutes and 45 minutes ahead of predictions. In other words, the model will predict on the next interval of 15 minutes, 30 minutes and 45. The data are divided into three parts: training, test and validation.

B. Multitask LSTM-NARK Architecture

The multi-task problems in this study are to predict tide level in four stations (locations) simultaneously. The first task or main task is to forecast the tidal level in Wick, and the other tasks (as auxiliary tasks) are to predict tidal in Workington (task 2), Ullapool (task 3) and New Heaven (task 4).

The prediction capability of the models was tested not only tested on validation and test data but also on other new tasks, including the tide level data at Barmouth, Cromer, Milford Haven and Tobermory.

C. Performance Metrics

Multiple loss functions were defined based on error values of the outputs of multiple tasks and the predicted results. STL may yield multiple local minimum values and the possibility that gradient vanishing appears in the training process is high. Because the tasks in MTL are executed simultaneously, the learning process can be prevented from falling into local minimal through sharing information between tasks [24].

An overall loss of the prediction model was obtained by summing losses of different tasks. The loss of task \( n \) is defined in the form of the root mean square error:

\[ L_{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} (\hat{y}_i - y_i)^2}{n}} \]  

(10)

where \( \hat{y}_i \) is the predicted value, \( y_i \) is actual value and \( n \) is the number of observations. The various task-specific loss functions must be combined into a single aggregated loss function by the joint loss function, which is defined as:
where $L$ is the joint loss and $m$ is the number of tasks.

### D. Results and Discussion

The optimal time lag in the multitask NARMAX – LSTM structure for each network were determined by using the lowest joint loss value obtained after performing empirical analysis. To find best neural networks architecture, the parameter settings for all the network models are as follows: the iteration number for each time horizon case is set to 200; the number of hidden layers, hidden nodes, and time lags for input and output are set to be from 1 to 5, 32 to 256, and 1 to 4, respectively.

The details of joint loss results for three different time horizon based on the best parameter setting are summarized in Table I, II and III.

It was observed from Table I that for one step ahead prediction, the optimal setting for the multi-task NARMAX-LSTM are as follows: time lag for input is $2$ ($n_u$), time lag for output is $1$ ($n_y$), time lag for noise is $1$ ($n_e$) and there are a total of $128$ nodes in the hidden layer.

The one step ahead forecast model based on the optimal model in Table I can be written as:

$$\hat{y}_1(t) = f_1(y_1(t-1), y_1(t-2), ... , y_4(t-1), y_4(t-2), u_1(t-1), ..., u_4(t-1), e_1(t-1), ..., e_4(t-1))$$  \hspace{1cm} (12)

For two step ahead case, the best model setting as follow: time lag for input is $4$ ($n_u$), time lag for output is $1$ ($n_y$), time lag for noise is $1$ ($n_e$) and there are $128$ nodes in the hidden layer as shown in Table I.

$$\hat{y}_1(t) = f_1(y_1(t-2), y_1(t-3), ... , y_4(t-2), y_4(t-3), u_1(t-1), u_1(t-2), u_1(t-3), u_1(t-4), ..., u_4(t-1), u_4(t-2), u_4(t-3), u_4(t-4), e_1(t-1), ..., e_4(t-1))$$  \hspace{1cm} (13)

<table>
<thead>
<tr>
<th>No.</th>
<th>Time Lag</th>
<th>LSTM Setting</th>
<th>Joint Loss (average RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$(n_y = 1, n_u = 1, n_e = 1)$</td>
<td>4 64</td>
<td>0.1542</td>
</tr>
<tr>
<td>2.</td>
<td>$(n_y = 1, n_u = 2, n_e = 1)$</td>
<td>1 128</td>
<td>0.1542</td>
</tr>
<tr>
<td>3.</td>
<td>$(n_y = 1, n_u = 3, n_e = 1)$</td>
<td>3 32</td>
<td>0.3185</td>
</tr>
<tr>
<td>4.</td>
<td>$(n_y = 1, n_u = 4, n_e = 1)$</td>
<td>1 32</td>
<td>0.2578</td>
</tr>
<tr>
<td>5.</td>
<td>$(n_y = 2, n_u = 1, n_e = 1)$</td>
<td>2 64</td>
<td>0.2416</td>
</tr>
<tr>
<td>6.</td>
<td>$(n_y = 2, n_u = 2, n_e = 1)$</td>
<td>1 32</td>
<td>0.1931</td>
</tr>
<tr>
<td>7.</td>
<td>$(n_y = 2, n_u = 3, n_e = 1)$</td>
<td>1 64</td>
<td>0.1774</td>
</tr>
<tr>
<td>8.</td>
<td>$(n_y = 2, n_u = 4, n_e = 1)$</td>
<td>2 64</td>
<td>0.2126</td>
</tr>
<tr>
<td>9.</td>
<td>$(n_y = 3, n_u = 1, n_e = 1)$</td>
<td>2 64</td>
<td>0.2248</td>
</tr>
<tr>
<td>10.</td>
<td>$(n_y = 3, n_u = 2, n_e = 1)$</td>
<td>2 32</td>
<td>0.2276</td>
</tr>
<tr>
<td>11.</td>
<td>$(n_y = 3, n_u = 3, n_e = 1)$</td>
<td>1 64</td>
<td>0.1854</td>
</tr>
<tr>
<td>12.</td>
<td>$(n_y = 3, n_u = 4, n_e = 1)$</td>
<td>1 64</td>
<td>0.1953</td>
</tr>
<tr>
<td>13.</td>
<td>$(n_y = 4, n_u = 1, n_e = 1)$</td>
<td>3 32</td>
<td>0.2988</td>
</tr>
<tr>
<td>14.</td>
<td>$(n_y = 4, n_u = 2, n_e = 1)$</td>
<td>1 32</td>
<td>0.2101</td>
</tr>
<tr>
<td>15.</td>
<td>$(n_y = 4, n_u = 3, n_e = 1)$</td>
<td>2 32</td>
<td>0.2380</td>
</tr>
<tr>
<td>16.</td>
<td>$(n_y = 4, n_u = 4, n_e = 1)$</td>
<td>3 64</td>
<td>0.2683</td>
</tr>
</tbody>
</table>

### TABLE II. TIME LAG PERFORMANCE FOR TWO STEPS AHEAD FORECASTING

<table>
<thead>
<tr>
<th>No.</th>
<th>Time Lag</th>
<th>LSTM Setting</th>
<th>Joint Loss (average RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$(n_y = 1, n_u = 1, n_e = 1)$</td>
<td>4 64</td>
<td>0.1287</td>
</tr>
<tr>
<td>2.</td>
<td>$(n_y = 1, n_u = 2, n_e = 1)$</td>
<td>1 32</td>
<td>0.1607</td>
</tr>
<tr>
<td>3.</td>
<td>$(n_y = 1, n_u = 3, n_e = 1)$</td>
<td>3 128</td>
<td>0.2273</td>
</tr>
<tr>
<td>4.</td>
<td>$(n_y = 1, n_u = 4, n_e = 1)$</td>
<td>1 128</td>
<td>0.1262</td>
</tr>
<tr>
<td>5.</td>
<td>$(n_y = 2, n_u = 1, n_e = 1)$</td>
<td>1 64</td>
<td>0.1607</td>
</tr>
<tr>
<td>6.</td>
<td>$(n_y = 2, n_u = 2, n_e = 1)$</td>
<td>3 256</td>
<td>0.2340</td>
</tr>
<tr>
<td>7.</td>
<td>$(n_y = 2, n_u = 3, n_e = 1)$</td>
<td>1 32</td>
<td>0.1798</td>
</tr>
<tr>
<td>8.</td>
<td>$(n_y = 2, n_u = 4, n_e = 1)$</td>
<td>4 64</td>
<td>0.2326</td>
</tr>
<tr>
<td>9.</td>
<td>$(n_y = 3, n_u = 1, n_e = 1)$</td>
<td>3 32</td>
<td>0.2697</td>
</tr>
<tr>
<td>10.</td>
<td>$(n_y = 3, n_u = 2, n_e = 1)$</td>
<td>1 32</td>
<td>0.1633</td>
</tr>
<tr>
<td>11.</td>
<td>$(n_y = 3, n_u = 3, n_e = 1)$</td>
<td>1 64</td>
<td>0.1693</td>
</tr>
<tr>
<td>12.</td>
<td>$(n_y = 3, n_u = 4, n_e = 1)$</td>
<td>1 64</td>
<td>0.1621</td>
</tr>
<tr>
<td>13.</td>
<td>$(n_y = 4, n_u = 1, n_e = 1)$</td>
<td>4 64</td>
<td>0.5914</td>
</tr>
<tr>
<td>14.</td>
<td>$(n_y = 4, n_u = 2, n_e = 1)$</td>
<td>2 64</td>
<td>0.2340</td>
</tr>
<tr>
<td>15.</td>
<td>$(n_y = 4, n_u = 3, n_e = 1)$</td>
<td>2 128</td>
<td>0.2083</td>
</tr>
<tr>
<td>16.</td>
<td>$(n_y = 4, n_u = 4, n_e = 1)$</td>
<td>1 128</td>
<td>0.1731</td>
</tr>
</tbody>
</table>

### TABLE III. TIME LAG PERFORMANCE FOR THREE STEPS AHEAD FORECASTING

<table>
<thead>
<tr>
<th>No.</th>
<th>Time Lag</th>
<th>LSTM Setting</th>
<th>Joint Loss (average RMSE)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>$(n_y = 1, n_u = 1, n_e = 1)$</td>
<td>1 64</td>
<td>0.1739</td>
</tr>
<tr>
<td>2.</td>
<td>$(n_y = 1, n_u = 2, n_e = 1)$</td>
<td>2 32</td>
<td>0.2317</td>
</tr>
<tr>
<td>3.</td>
<td>$(n_y = 1, n_u = 3, n_e = 1)$</td>
<td>2 32</td>
<td>0.2252</td>
</tr>
<tr>
<td>4.</td>
<td>$(n_y = 1, n_u = 4, n_e = 1)$</td>
<td>2 128</td>
<td>0.1929</td>
</tr>
<tr>
<td>5.</td>
<td>$(n_y = 2, n_u = 1, n_e = 1)$</td>
<td>5 64</td>
<td>0.2767</td>
</tr>
<tr>
<td>6.</td>
<td>$(n_y = 2, n_u = 2, n_e = 1)$</td>
<td>1 64</td>
<td>0.1546</td>
</tr>
<tr>
<td>7.</td>
<td>$(n_y = 2, n_u = 3, n_e = 1)$</td>
<td>3 128</td>
<td>0.2427</td>
</tr>
<tr>
<td>8.</td>
<td>$(n_y = 2, n_u = 4, n_e = 1)$</td>
<td>1 256</td>
<td>0.1566</td>
</tr>
<tr>
<td>9.</td>
<td>$(n_y = 3, n_u = 1, n_e = 1)$</td>
<td>3 64</td>
<td>0.2723</td>
</tr>
<tr>
<td>10.</td>
<td>$(n_y = 3, n_u = 2, n_e = 1)$</td>
<td>1 32</td>
<td>0.2037</td>
</tr>
<tr>
<td>11.</td>
<td>$(n_y = 3, n_u = 3, n_e = 1)$</td>
<td>2 256</td>
<td>0.1802</td>
</tr>
<tr>
<td>12.</td>
<td>$(n_y = 3, n_u = 4, n_e = 1)$</td>
<td>2 128</td>
<td>0.2324</td>
</tr>
<tr>
<td>13.</td>
<td>$(n_y = 4, n_u = 1, n_e = 1)$</td>
<td>1 32</td>
<td>0.2733</td>
</tr>
<tr>
<td>14.</td>
<td>$(n_y = 4, n_u = 2, n_e = 1)$</td>
<td>1 128</td>
<td>0.1723</td>
</tr>
<tr>
<td>15.</td>
<td>$(n_y = 4, n_u = 3, n_e = 1)$</td>
<td>3 64</td>
<td>0.2985</td>
</tr>
<tr>
<td>16.</td>
<td>$(n_y = 4, n_u = 4, n_e = 1)$</td>
<td>3 64</td>
<td>0.2596</td>
</tr>
</tbody>
</table>

As shown in Table III, the best model for the three steps ahead horizon case is as follows: two lagged input variables, two lagged output variables, one lagged noise variable and 64 nodes in hidden layer. For three steps ahead prediction, the model is:

$$\hat{y}_1(t) = f_1(y_1(t-3), y_1(t-4), y_1(t-5), ..., y_4(t-3), y_4(t-4), y_4(t-5), u_1(t-1), u_1(t-2), ..., u_4(t-1), u_4(t-2), e_1(t-1), ..., e_4(t-1))$$  \hspace{1cm} (13)

By analyzing the time lag variation and the LSTM setting from three different time horizons in Table I, II and III, we can see that the numbers of input time lag, output time lag and number of hidden nodes are highly related to the performance of the model.
time lag, hidden layers and hidden nodes have no correlations at all. The increase of the network complexity, namely, the increase of hidden layers, number of nodes and time lag does not have significant effect on the accuracy or joint loss value of the models. In some cases, it needs a relatively larger number of hidden layers to generate the lowest joint loss value while in other cases it only needs a single hidden layer with a minimal number of hidden nodes. This means that manual intervention is still needed in the fine-tune process of the neural network models.

The models for the above three different time horizons were tested using four new tasks, their performance were analyzed and summarized as follows. As can be seen in Table IV, individual loss values of the tasks (old and new) in a same time horizon are slightly different from each other. No task has an obvious larger loss value than other tasks. This means that the performance of some specific tasks does not significantly impact the performance of the models for other tasks. The joint loss value results show that both new and old tasks do not encounter overfitting.

**TABLE IV. COMPARISON OF INDIVIDUAL LOSS VALUE OF OLD TASKS AND NEW TASKS**

<table>
<thead>
<tr>
<th>Horizon</th>
<th>Task 1</th>
<th>Task 2</th>
<th>Task 3</th>
<th>Task 4</th>
<th>Joint Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0981</td>
<td>0.2167</td>
<td>0.1492</td>
<td>0.1527</td>
<td>0.1542</td>
</tr>
<tr>
<td>2</td>
<td>0.1228</td>
<td>0.1356</td>
<td>0.1136</td>
<td>0.1331</td>
<td>0.1262</td>
</tr>
<tr>
<td>3</td>
<td>0.0973</td>
<td>0.2026</td>
<td>0.1716</td>
<td>0.1472</td>
<td>0.1546</td>
</tr>
</tbody>
</table>

To further assess the capability of the proposed multi-task NARMAX-LSTM model, its performance was compared with LSTM, ARIMA and Backpropagation models. The comparison results are shown in Figure 2, from which it is clear that in term of joint loss, multi-task NARMAX LSTM outperforms all the other models in all task in three different time horizon.

From Figure 2, it can be seen that ARIMA severely suffers negative transfer: its RMSE values for Tasks 1 and 4 are much higher than those of the other models. The ARIMA model also shows the worst performance for all tasks and for all time horizons. Backpropagation also suffers negative transfer in all time horizons. The negative transfer occurs in Task 2 and Task 4 for the Backpropagation model.

The LSTM model performs good in one step ahead especially in task 1, task 2 and task 3 but have poor results in other time horizons.

V. CONCLUSION AND FUTURE WORKS

The paper proposed a novel multi-task learning approach for time series forecasting by effectively integrating the NARMAX and LSTM models. Unlike traditional time series forecasting methods which follow ‘a single model for a single task’ practice, the proposed MLT-NARMAX-LSTM method deals with time series modelling and forecasting from a multi-task viewpoint: it does not treat a single time series modelling and forecasting as an isolated task, but one of a set of tasks; the individual tasks are not the same but closely associated to each other (e.g. common features, similar change patterns, sharable knowledge and information). The proposed method has several advantages, for example, it generates robust models by making use of information shared by all the tasks; it avoids overfitting that can easily occur when using traditional time series modelling methods; the combination of the two different types of modelling methods, namely, NARMAX and LSTM, makes the proposed modelling framework immune to negative transfer. The predictive capability of MTL-NARMAX-LSTM network models was tested and evaluated on several real datasets of tide levels. Experimental results confirm the good performance of the proposed method.

In future, more benchmarks and case studies for applications in different areas will be investigated to further explore and improve the performance of the proposed method. To explore a new or improved MTL framework by employing evolutionary computation methods and transfer learning approach will be another interesting direction.

![Figure 2](image-url)

Figure 2. Comparison of the joint loss of the NARMAX-LSTM, LSTM, Backpropagation, and ARIMA. (a) One Step Ahead (b) Two Steps Ahead (c) Three Steps Ahead

ACKNOWLEDGMENT

The authors acknowledge that this work was supported in part by NERC (Ref. NE/V001787/1), NERC (Ref. NE/V002511/1) EPSRC (Ref. EP/H00453X/1), EPSRC (Ref. EP/I011056/1), EU Horizon 2020 (Ref. 637302). Nerfita Nikentari gratefully acknowledges a scholarship from the Indoensia Endowment Fund for Education/Lembaga Pengelola Dana Pendidikan (LPDP).
REFERENCES


Formation Drilling RoP Prediction via Deep Neural Networks with TensorFlow

Sherrif Isam
Department of Computer Science, Faculty of Engineering and Informatics, University of Bradford, Bradford, West Yorkshire, BD7 1DP, UK
s.isam@bradford.ac.uk

Qichun Zhang
Department of Computer Science, Faculty of Engineering and Informatics, University of Bradford, Bradford, West Yorkshire, BD7 1DP, UK
q.zhang17@bradford.ac.uk

Abstract—Predicting the speed for the drill penetrates the formation (RoP, Rate of Penetration) has been considered as an essential factor for the drilling operation and efficiency. Complexity in predicting the RoP arises from its dependency on many factors, for example, drilling fluid properties, drilling parameters, and the characteristics of the drilled formation. Based on the mentioned factors, the objective of this paper is to use Deep Neural Networks (DNN) with TensorFlow to predict RoP based on the 8 distinct parameters, which are obtained in the Final Well Report-Well 15/9-F-15 of the Volve oil field data provided by Equinor. The obtained results confirmed that the three DNN Models with TensorFlow techniques are applied to estimate the complex lithologies RoP with varying accuracies. As a comparative study, the presented Model 3 in this paper which includes an input layer with 32 neurons, 5 deep layers with 64 neurons, and a single neuron output layer has outperformed the previous 2 smaller models and it has been recorded how all the performance parameters has recorded smaller error values as the models adopted new deep layer and neurons.

Keywords—Rate of penetration, formation drilling, deep neural networks, TensorFlow

I. INTRODUCTION

Drilling is a complex operation consisting of variety of processes that require well planned infrastructure and well-trained specialists who are required to adhere strictly to individual states regulations [3]. Rotary drilling is considered as the common approach used for drilling operations, where the rig is in constant operation for approximately 21 to 28 days [4].

A typical well is drilled in several stages. Drilling starts with large diameter drill bits and successfully bring in smaller bits as the drilling is advanced. The speed when the bit breaks the rock (RoP, usually measured in meters per hour) has been the subject for many studies and optimizing it should bring in significant benefits in time and cost requirements [10]. A non-hazardous mixture based on bentonite clay or synthetic thickeners called the drilling mud can be pumped via the pipe and circulated to the surface will lift rock cuttings to the surface in addition to helps to stabilize the hole, cool the drill bit, and control downhole pressure. After drilling each portion of the well, a nested steel protective casing is cemented into place to protect ground water and maintain the integrity of the well [2]. Figure 1, recalled in [2], shows a typical schematic.

Optimization of drilling is a topic that is numerous researched, Eren describes optimized drilling as: “a system of pre-selecting the magnitude of controllable drilling variables to maximize footage or minimize drilling cost” in [13]. [12] identified two classes of operating parameters, controllable and uncontrollable, that can be used to produce a convenient estimate the RoP. The parameters have been illustrated in Figure 2 [6]. The large number of parameters affecting RoP gives rise to uncertainty and non-linearity, and it is the reason behind not having an exact mathematical model describing RoP [21].

Fig. 1. Schematic showing the main parts of a rotary drilling rig

Fig. 2. Schematic showing Drilling parameters and their relation to RoP
While reviewing the literature, it has been discovered that there are seven main RoP management approaches: Analytical, Semi-Analytical, Empirical, Managed Pressure Drilling, Artificial Intelligence Algorithms and Bit Management [19]. Different models have also been developed for the two different bit types Roller Cone Barrel (RCB) and Polycrystalline Diamond Compact (PDC) due to their fundamental differences. The most significant analytical model is: Bingham Model (1964), it relates RoP to formation properties, bit properties, rotation per minute (RPM) and weight on bit (WOB) and Bourgoyne and Young’s (1974) popular relation for a RCB which relating RoP to formation properties, formation compaction, pressure overbalance, bit weight, tooth wear, rotary speed, and bit hydraulics [19]. In 1987, Warren provided his two mode (perfect and imperfect hole cleaning) RoP model that modifies the jet impact force \( F_{jm} \) and includes the confined compressive strength, bit diameter, RPM, WOB, plastic viscosity and fluid specific gravity of the drilling fluid [19]. Hereland and Hoberock (1993) upgraded Warren’s model by adding the bit wear function \( W \) and the “chip hold-down function” [19]. Rastegar (2008) has further modified Hereland and Hoberock by a physics-based equation and Al-abduljabbar (2019) introduced a correlation using regression analysis [19]. Several empirical correlations are also developed for RoP prediction; beside drilling parameters, Duket and Bates (1980) considered effective formation strength while Seifabad and Ehteshami (2013) have considered the effect of the geological data from 50 wells in the Ahvaz oil field [22]. [5] have used data from 25 wells from Mishrif Southern Iraq to enhance RoP by utilising statistical analysis approach (SAA) and mechanical specific energy (MSE). Even though, laboratory studies have confirmed the aforementioned and other traditional physics based models to be reliable but there was always a conditions and assumptions with regards to perfect hole cleaning and constants to be calculated either from field readings or other experimental methods, this resulted in conventional methods to be both time consuming and was hard to generalize for the entire well [20] and [14].

Like in many other engineering fields, big data infrastructure, abundance of real time data have made it possible for engineers to deeply analyze data and use machine learning to compute non-linearity. Nowadays, AI is used in all domains in oil and gas industry but by far the most popular are: ANN, fuzzy logic and genetic algorithms, other techniques that are also widely used are support vector machine (SVM) and cased based reasoning (CBR) [7].

Both Bilgesu and Elkatatny have used ANN to predict RoP, Bilesu has implemented it in data from 9 formations and Elkatatny’s model achieved 4% absolute percentage error in predicting 2700 unseen RoP values [12]. [18] has shown significant accuracy in predicting RoP for North Sea horizontal wells using recent AI technologies with statistical regression coupling [18]. Implementing SVM, [1] has outperformed all RoP theoretical equations achieving an absolute percentage error of 2.83%, and a correlation coefficient of 0.997. [6] proposed a hybrid-data driven optimization system resulting in a faster RoP with 22% and 15-20% improvements in efficiency and cost reduction, respectively. [24] has shown that when there is large amount of data available, Adaptive-Neuro-Fuzzy-Inference-System (ANFIS) achieves better performance than the conventional models of Bourgoyne and Young (BY) and Hareland-Rampersad (HR).

[16] used popular metaheuristic algorithms to compare two methods (eyeball and random-search) in evaluating trade-off between searching for global RoP optimal and the computational run time, they claimed that the investigated data-driven models were adopted for real-time drilling with computational constraints.

II. DATA DESCRIPTION

The data used in this study is transmitted via WITSML from the Volve oil field which is located at the Southern end of the Norwegian sector. In particular, the location is shown by Figure 3 [23]. The field was discovered in 1993 and shutdown in 2016 after producing 10 million cubic meters of oil [11].

All of the input data is extracted from the Mud Log Report prepared by Geoservices for StatoilHydro [15].

Fig. 3. Schematic showing Volve Oil Field

A. Wells Information

The project development under investigation included drilling of the main exploration well F-15 middle of the southern segment of the field and three side-tracks – F-15 A, F-15 B, and F-15 C. In this project only data from the main well (F-15) is used for modelling.

Mudlogging equipment: Geoservices ALS3b (v1.09b) software was used for this well and real time depth data was collected at 1m intervals throughout the well [15].

B. Drill Bit, Mud Type and Lithology

Since several studies revealed different results for modelling wells drilled with different drill bits, the table below shows the drill bits used in this project, the mud type used, and the lithology drilled through.
TABLE I. SHOWS DRILL BIT, MUD AND LITHOLOGY TYPES

<table>
<thead>
<tr>
<th>Hole Section</th>
<th>BHA Number</th>
<th>Bit type</th>
<th>Depth in (m)</th>
<th>Depth Out</th>
<th>Mud Type</th>
<th>Lithology</th>
</tr>
</thead>
<tbody>
<tr>
<td>36”</td>
<td>1</td>
<td>XR-ODVC</td>
<td>144</td>
<td>226</td>
<td>Seawater</td>
<td>Returns to seabed</td>
</tr>
<tr>
<td>26”</td>
<td>2</td>
<td>XR-VC</td>
<td>226</td>
<td>1378</td>
<td>Spud Mud</td>
<td>Returns to seabed</td>
</tr>
<tr>
<td>17 1/2”</td>
<td>3 (F15)</td>
<td>CL423</td>
<td>1378</td>
<td>1381</td>
<td>OBM</td>
<td>Claystone</td>
</tr>
<tr>
<td>12 1/4”</td>
<td>4 (F15)</td>
<td>RS616M-A12</td>
<td>1381</td>
<td>2536</td>
<td>OBM</td>
<td>Clayst., sandst.</td>
</tr>
<tr>
<td>8 1/2”</td>
<td>5 (F15)</td>
<td>MDI 716</td>
<td>2536</td>
<td>3670</td>
<td>OBM</td>
<td>Limestone</td>
</tr>
<tr>
<td>8 1/2”</td>
<td>6 (F15)</td>
<td>MDI 616 LGBPX</td>
<td>3670</td>
<td>4090</td>
<td>OBM</td>
<td>Clayst., Sandst., Limest</td>
</tr>
<tr>
<td>17 1/2”</td>
<td>7 (F-15 A)</td>
<td>Unknown</td>
<td>1381</td>
<td>2591</td>
<td>OBM</td>
<td>Claystone</td>
</tr>
<tr>
<td>17 1/2”</td>
<td>8 (F-15)</td>
<td>RSR716S-D1</td>
<td>2591</td>
<td>2593</td>
<td>OBM</td>
<td>Claystone</td>
</tr>
<tr>
<td>8 1/2”</td>
<td>9 (F-15 A)</td>
<td>MDI716</td>
<td>2593</td>
<td>2596</td>
<td>OBM</td>
<td>Clayst., Sandst., Limest</td>
</tr>
<tr>
<td>8 1/2”</td>
<td>10 (F-15 A)</td>
<td>MDI716</td>
<td>2596</td>
<td>3180</td>
<td>OBM</td>
<td>Clayst., Sandst., Limest</td>
</tr>
<tr>
<td>8 1/2”</td>
<td>11 (F-15 A)</td>
<td>MDI 616LGBPX</td>
<td>3180</td>
<td>4095</td>
<td>OBM</td>
<td>Clayst., Sandst., Limest</td>
</tr>
<tr>
<td>8 1/2”</td>
<td>12 (F-15 B)</td>
<td>pilot MDI 616LGBPX</td>
<td>3185</td>
<td>3497</td>
<td>OBM</td>
<td>Clayst., Sandst., Limest</td>
</tr>
<tr>
<td>12 1/4”</td>
<td>13 (F-15 C)</td>
<td>MDI716</td>
<td>2562</td>
<td>2665</td>
<td>OBM</td>
<td>Clayst., Sandst., Limest</td>
</tr>
<tr>
<td>12 1/4”</td>
<td>14 (F-15 C)</td>
<td>MDI716</td>
<td>2665</td>
<td>2920</td>
<td>OBM</td>
<td>Clayst., Sandst., Limest</td>
</tr>
<tr>
<td>8 1/2”</td>
<td>15 (F-15 C)</td>
<td>MDI616LGBPX</td>
<td>2920</td>
<td>3232</td>
<td>OBM</td>
<td>Clayst., Sandst., Limest</td>
</tr>
</tbody>
</table>

III. METHODOLOGY

The workflow of this paper is illustrated by figure 4. Data is sent to the central processing office via WITSML and AZURE Storage Explorer is used to explore data. Data used here is extracted from the final mud log report. [15] illustrates the main well data frame, it is mentioned that measurements are averaged over 5 meters interval of Total Depth.

Fig. 4. The workflow of the approach in this paper

Data is read into panda for processing, total vertical depth (TVD) and drillability exponent (D Exp) columns are removed because the total depth is the chosen parameter in this study to represent depth along the well bore and D Exp is excluded as it is an extrapolation of other drilling data (such as RoP and Mud Weight) which are already included in the modelling. The remaining data frame, now has a shape of 789 by 9 is split into 60% training data (shape = (473,9)) and 40% temporary test data (shape = (316,9)) using train_test_split imported from model_selection in the sklearn library. Temporary test data is further split via aforementioned method into 50% test data set and 50% validation data set.

During data preprocessing scatter plots of all numerical attributes against each other were constructed. The heat map in Fig 5 summaries all the correlations between the features included in modelling.

From the heat map there is a strong collinearity between SPP vs MW IN, Torque vs Depth. The heat map also shows that RoP is in negative correlation with all the features except for Weight on Bit (WOB) and FLWpmps, but it is apparent that the strongest negative correlation with RoP is with MW IN (-0.43) and Depth (-0.46). Because the aim of this paper is to investigate the ability of Neural Networks to predict RoP, all features will be included in the modelling while the relation of specific features to RoP maybe investigated in later studies.

Fig.5. Features Correlation

The labels (RoP) are extracted from the three data sets using the “pop” method before normalising train, test and validation data sets via the following function:

```python
def norm(x):
    return (x- train_stats['mean'])/train_stats['std ']

normed_train_data = norm(train_dataset)
normed_test_data = norm(test_dataset)
normed_valid_dataset = norm(valid_dataset)
```

Three DNN models using TensorFlow are built parallel to each other, entitled Model 1, Model 2 and Model 3, where the structures are indicated below. The input shape for all three models was explicitly designated and all hidden layers were “relu” activated. Only one optimizer is used for all models which is “Adam” at a learning rate of 0.001. All models were compiled to: loss = ‘mse' and
metrics = ['mae', 'mse', 'mape']. Before training the model, check points were saved during training using keras through declaring a callback function to be called at the end of each epoch which picks the best model (model is at a local optima and with good convergence) based on the validation loss.

The presented model has been trained using the normalised data set and training labels using a Tesla K80 GPU, a 2.8.0 TensorFlow version and Python 3.7.12. For memory management reasons and the ability of neural networks to take the data in small chunks, input data is fed in batch size of 32.

In particular, we have

<table>
<thead>
<tr>
<th>Table II: A SUMMARY OF MODEL 1, 2, AND 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Model 1</td>
</tr>
<tr>
<td>Model 2</td>
</tr>
<tr>
<td>Model 3</td>
</tr>
</tbody>
</table>

Figs. 6 to 8 illustrate MAE and MSE curves of training vs validation data sets for all three models in addition to plots to true labels versus predicted labels on the training and test data sets. It can be observed from figs. 6-8 that in all three models the loss functions have decreased indicating the models were learning but the overfitting was apparent on the validation data sets.

### IV. RESULTS

In this paper, artificial neural networks has been adopted to predict the rate of penetration using total depth, WOB, torque, RPM, Standpipe Pressure (SPP), mud flow rate, total gas and mud weight. Three models have been trained with various deep layers and neurons but the same activation and loss functions. The results (Table II) shows that as the number of neurons and deep layers increased the model performance has improved.

<table>
<thead>
<tr>
<th>TABLE III. SHOWS RESULTS FROM THREE MODELS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>Model 1</td>
</tr>
<tr>
<td>Model 2</td>
</tr>
<tr>
<td>Model 3</td>
</tr>
</tbody>
</table>

In summary, the following conclusions are drawn based upon the modelling results:

1. Deep neural networks with multiple layers were able to predict RoP to certain accuracy using raw data.
2. As the structure of deep layers with the number of neurons increased modelling accuracy has the potential to be improved.

Such an approach requires all RoP data to be available during training. This obviously will mean that the well has been drilled and data is collected. A greater contribution will be to predict the RoP while drilling. To do this we could use well log data and RoP from previously drilled exploration well and try predicting RoP in newer production wells. Also, as a future study, neuro-network system [25] and wavelet packet transform [26-29] can be used to apply human-like reasoning through the use of fuzzy sets and rules based on type of lithology the bit is drilling through.

Figs. 6 to 8 illustrate MAE and MSE curves of training vs validation data sets for all three models in addition to plots to true labels versus predicted labels on the training and test data sets. It can be observed from figs. 6-8 that in all three models the loss functions have decreased indicating the models were learning but the overfitting was apparent on the validation data sets.
ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments.

REFERENCES


An Introductory Survey of Entropy
Applications to Information Theory, Queuing Theory, Engineering, Computer Science, and Statistical Mechanics

Ismail A Mageed1 Qichun Zhang2
Department of Computer Science, Faculty of Engineering and Informatics, University of Bradford, Bradford, West Yorkshire, BD7 1DP, UK
1iammoham@bradford.ac.uk 2Q.Zhang17@bradford.ac.uk

Abstract— Clausius was the first to coin the term “entropy” roughly 160 years ago. Since then, many scholars from several scientific fields have continued to enhance, develop, and interpret the data. This study describes the concept of entropy and its applications to information theory, queuing theory, engineering systems, computing, and statistical mechanics. The choice of these applications was subjected to the size limitation of the paper. Having said that, entropy has a great applicability to other scientific disciplines. Even more interesting is the nature of entropy as a measurement of uncertainty, disorder, or turbulence was the main core of attraction to a considerable number of scholars in all academic areas of research expertise to both employ it and make a significant research advancement. Its goal is to present a coherent picture of how classical entropies might be used in engineering and computer science. It is attempted to create a concise exposition that strikes a balance between conceptual importance and instrumental relevance. A representative list of references closes this review.

Keywords—Survey, entropy, information theory, queueing theory, engineering systems, computing, statistical mechanics.

I. INTRODUCTION

It is well agreed that entropy is a broad notion to be found in various frameworks with completely different interpretations. Therefore, there are many faces for entropy, it could be the most celebrated Boltzmann-Gibbs entropy that measures the microscopic disorder within a statistical mechanic framework; the information-theoretic Shannonian entropy functional measuring both uncertainty and compression. Another face is the measurement of uncertainties and randomness in dynamical complex systems / ergodic theory and continuous dynamics by employing the famous Kolmogorov-Sinai entropy functional. Yet, comes another measure of complexity in continuous dynamics, namely the topological entropy. As known in contemporary science, entropy the giant upon which shoulder, the second law of thermodynamics will mount to be associated by most relativists to be coined with the arrow of time.

It is to be noted that entropy is to be found right at the core of all contemporary scientific disciplines, for example, cryptography, coding theory, information theory and much more. Due to the universality of entropic applicability, it becomes so difficult to encounter the huge number of applications of entropy in science. To see this more closely, the Lempel-Ziv information-theoretic complexity, the computational algorithmic complexity and much more. The axiomatic uniqueness of the entropic functional was characterized by Shannon [1]. Following the footsteps of Shannon’s discovery, Rényi [2], Havrda-Charvat [3], and Tsallis [4] came up with generalizations to Shannon’s entropy with much weakened conditions or different axiomatic approaches. The generalized Rényian entropy framework has uncovered new avenues to devise many generalized entropies to extend the applicability to the underlying physics associated to complex systems [5]. Historically speaking, a big circle of ideas has been obtained as a brainchild of combining the trio: information theory with statistical mechanics and complex dynamics. Regarding to mathematical and physical fields, the theory of system dynamics is originated from statistical mechanics, from which it inherited the main conceptualization of ergodicity, equilibrium state, invariant measure, orbit, stationary state, and entropy.

The Shannonian conceptualization of information theory was purely a probabilistic framework. Information sources are stationary random processes. In statistical mechanics, an information-theoretic approach in the form of is needed to solve the most celebrated Maxwell’s paradox [6]. Recently, more novel entropies have appeared [7-13]. These are regarded as entropy-like quantities, which are primarily uncovered new avenues to devise many generalized entropies to extend the applicability to the underlying physics associated to complex systems [5]. Historically speaking, a big circle of ideas has been obtained as a brainchild of combining the trio: information theory with statistical mechanics and complex dynamics. Regarding to mathematical and physical fields, the theory of system dynamics is originated from statistical mechanics, from which it inherited the main conceptualization of ergodicity, equilibrium state, invariant measure, orbit, stationary state, and entropy.

The Shannonian conceptualization of information theory was purely a probabilistic framework. Information sources are stationary random processes. In statistical mechanics, an information-theoretic approach in the form of is needed to solve the most celebrated Maxwell’s paradox [6]. Recently, more novel entropies have appeared [7-13]. These are regarded as entropy-like quantities, which are primarily obtained to solve the challenging problems in chaos, multiscale analysis, cellular automata and much more. Alongside with these proposed entropies, classic entropies are still possessing a great scope of applicability in ergodic theory.

![Figure 1. Transfer entropy $T_{c,m-1,m}$ for the coupling direction as a function of the coupling strength $\epsilon$ in a tent map lattice (binary partition). (c.f., [13])](image-url)
Figure 1 visualizes the numerical validity of these results for a spatially periodic 100-map grid. Also, it shows the average of 10 runs of $10^5$ iterations after the $10^5$ transient ones. The transfer entropy $T_{m\rightarrow m+1}$ as well as both directions $M (\tau = 1)$ have been found to match zero and are not displayed. In fact, entropy is unrivalled in mathematics, physical science, and social science in capturing elusive concepts such as disorder, information (or ignorance), randomness, complexity, and irregularity. For a more detailed account, the reader is advised to consult [14].

II. ENTROPY APPLICATIONS TO INFORMATION THEORY

Shannon [1] introduced two important information-theoretic properties, notably the information content and his well-known entropy. The information content is determined by

$$I = K \ln(M)$$

(1)

$I$ denotes the total information of a message, $K$ is a constant and $M$ is the number of probable strings in a finite set of information. The second property, entitled as entropy [1]. Shannon entropy $H$ defined as

$$H = -K \sum_n p(n) \ln p(n)$$

(2)

provided that function $p(n)$ is the probability of symbol $n$ in a string. Consequently, by using equations (1) and (2), we can interpret Shannon entropy as the negative expected value of total information of a message having $p(n)$ is the probability of symbol $n$ in a string.

It worth mentioning that negentropy is like information [15], which was the early spark to solve the unsolved problem of Maxwell demon. It was also declared that by using the research findings of Brillouin, irreversible logical operations [16], for example, erasing data imply physical irreversibility. Moreover, it is conjectured that thermodynamics entropy $S$ is equal to information $I$ [17]. In [18,19], a qualitative link between entropy and information was established. In other words, it has been proven in [18,19] that the maximum information about the system $I_{max}$ equals the subtraction of information necessary for its complete description from a finite information content. This has been investigated more thoroughly by [21-23,18] to obtain the equation governing a simple conservation law:

$$H + I = constant = H_{max} = I_{max}$$

(3)

$H$ serves as entropy, while $I$ denotes information, and $H_{max}$ and $I_{max}$ refer to the maximum possible values. This translates to Thus: "equal loss of entropy is always compensated by an equal loss of entropy" [18].

Figure 2. Information theory provides a quantitative interpretation of the distribution of stochastic liquidity. The topological space is divided into smaller cells of equal hypervolume, and it is assumed that all fluids are initially confined in one cell (a). The information required to specify this distribution is equal to the base 2 logarithm of the number of cells. As the particle system evolves, stochastic fluids gradually occupy more cells (b) (see [18]).

In practice, if $I_{max}$ reads as a certain information content of a book, then as we go through this book, then by the increase of the known information content $I$, the unknown information content, entropy $H$ decreases. As soon as the book is read, $I_{max}$ would be the maximum value attained by $I$. To see this more clearly, Information theory provides a quantitative interpretation of the distribution of stochastic liquidity, as clearly shown in Figure 2.

III. ENTROPY APPLICATIONS TO QUEUEING THEORY

It has been shown that the probability distribution changes over time for many stochastic processes. Consequently, the entropy of a probability distribution also changes over time. Given a stochastic distribution of the queueing system states, the corresponding value of the entropy can effectively be calculated to measure the level of the system state uncertainty. Mathematically speaking, this probability distribution is ambiguous to us. The derived information is explicitly formulated in parameters of the underlying queueing system such as mean arrival rate, etc.

An analytical exposition of the stable $M/G/1$ and $G/M/1$ queues and determining the corresponding service time distribution by employing entropy optimization has been undertaken by [24]. Moreover, using known mean values, a model by adopting maximum entropy conditions is determined [25]. It is to be noted that the resulting model was probabilistic.

In [26], a detailed account on how uncertainty is impacted by the variation of queueing parameters such as mean service rate, mean arrival rate and time. The results of this conducted research were very promising. The study of two single server discrete-time queues has been carried out by [27]. The authors of [27] have derived novel relationships with the time capacity of queueing systems.

![Figure 3. The queue-size process (c.f., [27])](image-url)

By observing that it arrives shortly after the start of the time slot and departs just before the end of the time slot. Therefore, there are always two components to the queue size $q_k$. The $q_{k-}$ measured shortly after the possibility of departure at the end of time $k-1$ and the $q_{k+}$ measured shortly after the possibility of arriving at the beginning of time $k$.This is captured visually in Figure 3.

There are many well-known measures of entropy in the information theory literature. All the entropy measurements are produced and motivated based on the most elementary Shannon entropy [1] as well as other available entropy measures within the literature [2,26,28-33]. These measures will facilitate a good contribution to study the uncertain behavior of queueing systems in different states as demonstrated by [34].

As a potential technique to investigate complex queueing models, the principle of maximum entropy (PME) appears
indifferent frameworks. PMEs are employed to obtain the probability of equilibrium subject to the first few moments as constraints of the underlying distribution along with using the principle of perfect convenience. As with the $M/M/1$ model, it is feasible to derive an accurate distribution of queue lengths. Numerous authors have contributed to studying a large number of queuing models using PME. PME measures uncertainty, previously introduced into information concepts with the help of Shannon [1] extended by [17]. In addition, [35] provided the axiomatic derivation of the maximum entropy imperative mood and the maximum cross entropy of the system modelling. [24] adopted $M/G/1$ and $G/M/1$ for PME in a balanced queuing system. Cantor [36] considered steady-state multi-server queues and provided tandem information-theoretic analysis based entirely on the queue's PME. [37,38] adopted PME to analyze the general queuing community. Kouvatos [39] discussed the queue length distribution of $G/G/1$ queues based entirely on PME in finite form. [25] provided a $M/G/1$ queuing devices probabilistic model whose use of PME challenges the expected customers number constraints which are introduced from the Pollaczek-Khinchine formula. [40] Calculated most of the entropy flow in the network. [41] talked about applying the maximum entropy to the moment problem. The maximum entropy analysis of the multi-server queue system model was developed by [42]. Interestingly, [43] is a new analysis frame based primarily on PMEs for stable $G/G/1$ queues that are in equilibrium under the constraints of the distribution of arrivals and service times. Provided the work. The optimal entropy analysis was introduced by the author of [44]. Recently, an analysis of the queue size distribution for the $G/G/1$ model was performed by [45] using PME. Many researchers have studied the $G/G/1$ model thoroughly but keeping in mind that the closed-form solution to the probability distribution of the number of customers in the system is an open problem.

III. ENTROPY APPLICATIONS TO ENGINEERING

Although the concept of entropy has been implemented in many areas [46], there are still unexplored areas that need to be investigated. Entropy has a variety of uses in industrial engineering. For example, problems have been resolved and entropy-based methods have been proposed for each problem. The difficulty of disposing of material handling systems in the context of automated guided vehicles (AGVs) in discrete component manufacturing systems [47] is investigated. Replenishment issues are like allocating available AGVs to meet relocation requirements and manage the efficient flow of parts in a factory. A complete entropy-based resource allocation algorithm was used, primarily based on the Kullback-Leibler-oriented divergence. It efficiently oversees the impact of the potential movements’ stability of the manufacturing unit prior to resource allocation [47]. The proposed algorithm is implementable for real-time cases. It attempts to balance the load for the manufacturing unit while meets the motion requirements generated by the schedule of factory operational areas. The 2nd issue [47] examines complete rankings and decisions, primarily based on average performance measurements. The combination of PME and Kullback-Leibler gave rise to the idea of divergence for developing a two-step algorithm to solve this problem. The proposed scheme relaxes the underlying population normality assumptions that constrain the frequency-theoretic algorithm, which does not need the prior distribution assumed via the Bayesian method. To do so, it contributes to the evaluation and selection of information. Their decision gives a measure of performance. Also, an entropy-based criterion for evaluating the two options. The rating [47] is entirely based on the difference in direction between the cumulative probability distributions of the introduced alternatives. With the help of [48], we have captured the causal relationships of complex structures by constructing an internal compositional alignment of temporal structures using a new information-theoretic measure, joint entropy. This is a sequence-based non-uniform membership measurement to conceptualize the inference of the uncertainty of the connection between two time series. It has been observed that joint entropy influences the analysis of Hénon maps. Various noises are introduced in the Hénon map to check the accuracy and sensitivity. Coupling entropy is also used to analyze the relationship between the US unemployment rate and the consumer price index (CPI), with CPI as the driving variable and unemployment as the response variable. In addition, [49] uses entropy-based iterative learning discrimination to characterize neural membrane interconnects. More interestingly, looking at figure 4, a binary tree showing the possible link topologies is portrayed.

![Figure 4. Binary tree displaying possible link "topologies" (c.f., [50])](image)

The pertinence of the notion of entropy in engineering design theory and methodology was once investigated with the aid of [50] by resorting to the notion of design complexity as visualized in Figure 5.

![Figure 5. The embodiment of concept 1(2PT-1PR) ROBOT (c.f., [50])](image)

IV. ENTROPY APPLICATIONS TO COMPUTER SCIENCE

Information theory is a mathematical science [51] dealing with the transmission of information throughout a noisy channel. It is widely agreed that the building block of information theory is mainly based on the concept of quantifying the amount of information in a message. Furthermore, entropy is the cornerstone for techniques including building decision trees, feature selection, fitting classification models and much more [51]. As already known by information theorists, it is obvious that calculating information as well as entropy is crucial in machine learning. Information theory deals with source coding. Moreover, it is concerned with error coding or channel coding [52]. The probabilistic instinct behind the quantification [53] of the