

# An Enhanced GRU Model With Application to Manipulator Trajectory Tracking

Zuyan Chen<sup>1</sup>, Jared Walters<sup>1</sup>, Gang Xiao<sup>2</sup>, Shuai Li<sup>1,\*</sup>

<sup>1</sup>College of Engineering, Swansea University, Swansea, U.K.

<sup>2</sup>Shenzhen In-cube Automation Co. Ltd.

## Abstract

Service robots, e.g. massage robots, have attracted more and more attention in recent years and the most popular study within this field is trajectory tracking. Due to the actual demand for service robots, the solution of trajectory tracking requires fast convergence and high accuracy. In order to solve the above issues, this paper proposed an enhanced Gated recurrent unit (GRU) to deal with trajectory tracking tasks of robot manipulators. The main feature of enhanced GRU is utilizing cell states as well as various gate units to build a novel neural cell. Besides, the presented enhanced GRU resolves the problem of the general neural network model and large memory occupancy. Then the derivations about the computational process of cell state and mixed hidden state of the proposed model have been illustrated. Finally, three trajectory tracking applications, comparison, and visual simulation have verified feasibility as well as the superiority of the enhanced GRU model.

Received on 22 November 2021; accepted on 23 December 2021; published on 07 January 2022

**Keywords:** Trajectory tracking, gated recurrent unit (GRU), neural hidden state, gate unit, robot manipulators.

Copyright © 2022 Zuyan Chen *et al.*, licensed to EAI. This is an open access article distributed under the terms of the [Creative Commons Attribution license](#), which permits unlimited use, distribution and reproduction in any medium so long as the original work is properly cited.

doi: 10.4108/airo.v1i.7

## 1. Introduction

Robot manipulator motion tracking, a fundamental subsidiary subject of robot motion planning control, has always drawn researchers' wide concern [4, 5]. As service robots, especially massage robot, attracts more and more attentions from public in recent years, the applications of manipulator motion tracking becomes more extensive [1]. For instance, as it is presented in Fig. 1, it is the class application of trajectory tracking in massage robot production [2]. Robot acquires trajectory information from historical memory, which could be last masseuse's technique or customer's usage trace. And then self-learning method is utilized to build motion imitation model. At last, massage robot gives a scheduled motion massage on costumers via pre-trained model. However, in order to improve the costumers' experience, trajectory tracking in massage robot requires real-time as well as accuracy more than other elements [3]. Therefore, a feasible solution

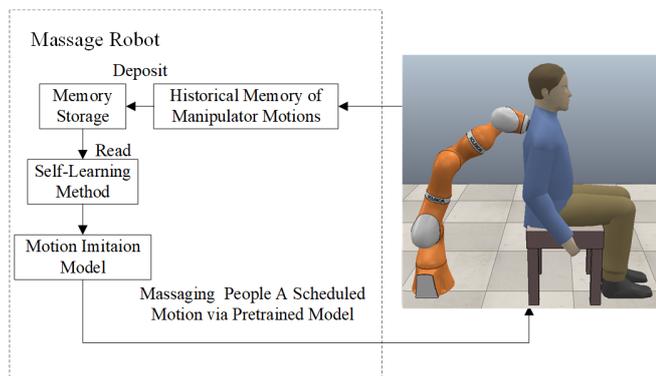
of manipulator trajectory tracking is urgent to be proposed.

So far, numerous excellent researchers have presented various high-performance solutions. For instances, A. Dumlu *et al.* [6] presented a fractional-order PID control to satisfy the high speed, high acceleration, and high accuracy control action need of parallel mechanisms. Besides, M. Galicki [7] derived a class of continuous Jacobian transpose robust controllers from non-singular terminal sliding vector variable and the Lyapunov stability theory to address the problem of finite-time convergence. A. Duka [8] solved the inverse kinematics by using a feed-forward neural network to generate desired trajectories in Cartesian space. In general, it is difficult to develop an effective and robust regular mathematics method because of singular point of solution as well as the high redundancy of cascade robot manipulator [9]. However, real-time motion tracking is diffusely applied in robotics field. Hence introducing a high-efficiency and short latency solution for tracking robot manipulator motion is the burning issue [10, 14].

\*Corresponding author. Email: [shuai.li@swansea.ac.uk](mailto:shuai.li@swansea.ac.uk).

**Table 1.** Features of Various Trajectory Tracking or Kinematics Solutions Application of Manipulator (KUKA LBR iiwa 7 R800)

Solution	Problem Solved	Model Size	Performance	Continuous	Prediction Speed
General IK Solution	Kinematics	Small	Good	No	Ordinary
Location Controller	Kinematics	Ordinary	Good	Yes	Ordinary
CNN	Patten Recognition	Small	Weak	No	Ordinary
RNN	Trajectory Tracking	Large	Good	Yes	Ordinary
LSTM	Trajectory Tracking	Large	Good	Yes	Ordinary
GRU	Trajectory Tracking	Small	Ordinary	Yes	Fast
Enhanced GRU	Trajectory Tracking	Ordinary	Good	Yes	Fast

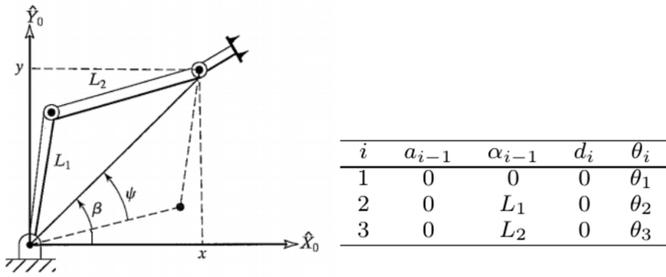


**Figure 1.** A classic application of motion tracking in massage robot production. Robot catches trajectory data from historical memory, which could be last masseuse's technique or customer's usage data. And then self-learning method is utilized to build motion imitation model. At last, massage robot gives a scheduled motion massage on costumers via pre-trained model.

Considerable approaches and technologies for tracking robot manipulator's motion have been comprehensively researched and introduced among the algorithms extended from general inverse kinematics [11], optimization theory [12], machine vision [13] as well as adaptive-control algorithms [10]. Traditional robot manipulator inverse kinematics solution has existed for several decades, but due to the multiple-solution and singular point, it could not perform well in continuous work. Over a span of past 10 years, machine learning has been the topic in focus of practitioners on account of its favourable performance and the convenient end-to-end modality. The most famous among these neural networks is convolutional neural network (CNN), however, majority of CNN models are applied in image processing [20]. In order to achieve the application in sequence processing, M. Wang *et al.* [15] proposed a novel CNN model (genCNN) with the ability of predict the next word with the history of words of variable length. Recurrent neural network (RNN), which was introduced in 1990 but attracted less interest from practitioners, had been refocused when it reached the

unexpected result in natural language processing [21]. Y. Li *et al.* [21] considered an advanced RNN structure to improve control precision and enhance adaptiveness for robot motion. Comparing with classical RNN models, long short-term memory (LSTM) could validly impede gradient explosion or disappearance and achieve better results by making well use of previous cell states. In [16], Sepp Hochreiter *et al.* first reported LSTM for resolving long time cost by storing information over extended time. With the introduction of forgot gate and sigmoid activation function, LSTM equips the ability to deal with long-term dependence. As LSTM gradually comes into researchers' attention, numerous developments based on LSTM have been studied. Gated Recurrent Unit (GRU) is one outstanding variation of LSTM, whose performance is similar to LSTM but with less computation [17]. Various novel works about RNN were published and presented. S. Li *et al.* [22], for instance, proposed a new RNN design to achieve efficient kinematic control of redundancy of manipulators in the presence of noises. P. Shrey *et al.* [18] introduced a robot learning from demonstration paradigm to imitate therapist's action based on LSTM. D. Robert *et al.* [19] compared four RNN architectures (simple RNNs, LSTM, GRU and mixed history RNNs) for recognizing complex action from kinematic of robot and indicated different performances of these models.

Although advanced techniques have tremendous achievements, the utilization of RNN in robot manipulator trajectory tracking reaches plateau in terms of large memory bandwidth, weakness for super long sequences as well as vast computation cost [23, 25, 34–36]. In fact, trajectory tracking requires Real-time, continuous and accuracy [24, 37, 38]. Therefore, this paper presented a novel enhanced GRU model to achieve the purpose of lower latency, smaller size, higher precision and continuous solutions in trajectory tracking tasks. For better understanding of advantages and disadvantages of various solution for robot manipulator trajectory tracking, comparisons are indicated in Table 1 [27–29]. Moreover, main innovation points as well as contribution of this paper are summarized as below.



**Figure 2.** It is a simple three connected rods architecture and its D-H parameters are presented, which is applied to explain how the general IK solutions work. Also, the mirror of the real pose indicates the disadvantage of general IK solution, termed Multiple-solution.

- This is an attempt to design robot manipulator trajectory tracking model by uniting cell state in GRU. The proposed enhanced GRU model is capable of less time to convergence as well as excellent prediction ability and solve the slightly low performance problem in GRU.
- The research bridge from various gate units to neural cell state as well as hidden state is triumphantly built, which provides more possibilities for researchers to develop more interesting studies above it.
- Complete formula derivation about enhanced GRU provides researchers convenience and clearer mind to improve the performance of enhanced GRU.

The following paper is arranged as below. Section 2, as research background, explains the principles of three inverse kinematic solutions. Section 3 details the training process and performance of enhanced GRU model. In Section 4, three trajectories-tracking applications, comparisons and simulations in V-rep are presented. Section 5 concludes conclusion of this study.

## 2. Related works

General inverse kinematics of solution processing with the three connected rods as example is firstly presented in this section. Afterwards, the principles and processing of the application of CNN, LSTM as well as GRU in trajectory tracking are indicated.

### 2.1. General Inverse Kinematics Solution

A three connected rods structure (D-H parameters are shown in Fig. 2) is utilized here for explaining how to resolve each joints' values from Transpose matrix [27]. Obviously, the kinematics formula of this equipment can be presented as below,

$${}^B_W T = {}^0_3 T = \begin{pmatrix} c_{1,2,3} & -s_{1,2,3} & 0 & l_1 c_1 + l_2 c_{1,2} \\ s_{1,2,3} & c_{1,2,3} & 0 & l_1 s_1 + l_2 s_{1,2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (1)$$

where  ${}^B_W T$  is the transpose matrix from rod B to rod W,  $c_{1,2,3}$  means  $\cos(\theta_1 + \theta_2 + \theta_3)$  and  $s_{1,2,3}$  stands for  $\sin(\theta_1 + \theta_2 + \theta_3)$ .

Then to simplify the operation process, some Algebra is assumed as below,

$$\begin{cases} c_\phi = c_{1,2,3} \\ s_\phi = s_{1,2,3} \\ x = l_1 c_1 + l_2 c_{1,2} \\ y = l_1 s_1 + l_2 s_{1,2} \end{cases} \quad (2)$$

Now algebraic method is applied to resolve the solution of (2). According to (2), we can acquire this formula,

$$x^2 + y^2 = l_1^2 + l_2^2 + 2l_1 l_2 c_2 \quad (3)$$

from 3,

$$c_2 = \frac{x^2 + y^2 - l_1^2 - l_2^2}{2l_1 l_2} \quad (4)$$

The condition that the above method has a solution is the right value of (4) must be  $-1$  to  $1$ . Physically, if (4) does not satisfy the situation, then the target location and pose is the destination that manipulator cannot reach. Supposing the target point is in the reachable space of robot manipulator, the expression of  $s_2$  should be,

$$s_2 = \pm \sqrt{1 - c_2^2} \quad (5)$$

At last, inverse tangent formula is utilized in (5),

$$\theta_2 = A \tan(s_2, c_2) \quad (6)$$

The  $\pm$  in 5 correspond multiple-solutions, which presents in this issue that the location of elbow of the manipulator (up or down). After getting the value of  $\theta_2$ , the value of  $\theta_1$  can be resolved connecting with (2). (2) can be transformed as below,

$$\begin{cases} x = k_1 c_1 - k_2 s_1 \\ y = k_1 s_1 - k_2 c_1 \\ k_1 = l_1 + l_2 c_2 \\ k_2 = l_2 s_2 \end{cases} \quad (7)$$

If we suppose  $r$  as follows,

$$r = |\sqrt{k_1^2 + k_2^2}| \quad (8)$$

and,

$$\gamma = A \tan(k_1, k_2) \quad (9)$$

then,

$$\begin{cases} k_1 = r \cos(\gamma) \\ k_2 = r \sin(\gamma) \end{cases} \quad (10)$$

so (7) could be changed as below,

$$\begin{cases} x/r = \cos(\gamma) \cos(\theta_1) - \sin(\gamma) \sin(\theta_1) \\ y/r = \cos(\gamma) \sin(\theta_1) - \sin(\gamma) \cos(\theta_1) \end{cases} \quad (11)$$

hence,

$$\begin{cases} \cos(\gamma + \theta_1) = x/r \\ \sin(\gamma + \theta_1) = y/r \end{cases} \quad (12)$$

applying double variables inverse tangent equation in (12),

$$\gamma + \theta_1 = A \tan(y/r, x/r) = A \tan(y, x) \quad (13)$$

at last,

$$\theta_1 = A \tan(y, x) - A \tan(k_2, k_1) \quad (14)$$

by combining the values of  $\theta_1$  and  $\theta_2$ , we can easily obtain the value of  $\theta_3$

## 2.2. Convolutional Neural Network

CNN, which are one of the representative algorithms of deep learning, are a variety of feed-forward neural networks which contain convolutional computation and with depth architecture [30]. The most essential convolutional computation process can be presented as below,

$$x_j^l = f\left(\sum_{i \in M_j} x_j^{l-1} k_{i,j}^l + b_j^l\right) \quad (15)$$

where  $x_j^l$  presents feature map  $j$  in the  $l$  layer,  $k_{i,j}^l$  indicates the convolutional kernel  $j$  in layer  $l$ ,  $b_j^l$  is the  $j$  bias in the  $l$  layer and function  $f$  stand for activation function. Tanh function is applied here, whose formula is shown as below,

$$f(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{1 - \exp(2x)}{1 + \exp(2x)} \quad (16)$$

as shown in (15), in order to determine the matrix value of  $k$  as well as  $b$ , gradient descent optimize algorithm is utilized here. Adam [31] is one of the commonly used adaptive algorithms among various optimization algorithms. The formula to calculate gradient as below,

$$g_t = \nabla_{\theta} f_t(\theta) \quad (17)$$

where  $g_t$  represents the gradient of  $f_t(\theta)$  while  $f_t(\theta)$  is the gradient of  $t$  epoch. The major step to achieve the goal is to decrease gradient as below,

$$\begin{cases} m_t = \beta_1 * m_{t-1} + (1 - \beta_1) * g_t \\ v_t = \beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2 \\ \alpha_t = \alpha \sqrt{1 - \beta_2^t} / (1 - \beta_1^t) \\ \theta_t \leftarrow \theta_{t-1} - \alpha_t m_t / (\sqrt{v_t} + \hat{\epsilon}) \end{cases} \quad (18)$$

where  $\beta_1$  and  $\beta_2$  are the number in  $[0,1)$  indicated by ourselves,  $m_t$  is the exponential mean while the  $v_t$  presents square gradient,  $\alpha_t$  indicates exponential attenuation controlled by parameters  $\beta_1$  as well as  $\beta_2$ .

## 2.3. Gated Recurrent Unit

As one popular variates of LSTM, GRU is introduced for using less computation resources to reach the similar performance of LSTM. Compared with LSTM containing three gates, GRU has simpler structure with only two gates, update gate as well as reset gate [33]. The expression of GRU is as below,

$$\begin{cases} r_t = \sigma(W_r * [h_{t-1}, x_t]) \\ z_t = \sigma(W_z * [h_{t-1}, x_t]) \\ \tilde{h}_t = \tanh(W_{\tilde{h}} * [r_t * h_{t-1}, x_t]) \\ h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \\ y_t = \sigma(W_o \cdot h_t) \end{cases} \quad (19)$$

the  $r_t$  in 19 is belonged to reset gate, which is applied to control the extent to which the previous cell state affect current cell state. The function of  $z_t$  in update gate is to forget or choose to memorize information from last cell.

## 2.4. Cell State and Gate Units

Cell state and Gate Units are introduced to solve the Long-time dependency problem [32]. Cell state could be deemed to store historical information and record the last cell's situation. There is still the issue that a part of invalid information exists in the cell state and it is why gated units were presented to alleviate this problem. The most outstanding gate units are forgotten gate, input gate as well as output gate. The first gate named forgotten gate, which is utilized to drop information in cell state. The formula is presented as below,

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (20)$$

where the  $f_t$  shows the percentages of retained information in previous cell state, the  $h_{t-1}$  presents the hidden state of cell, and the  $\sigma$  in 20 stands for sigmoid function,

$$\sigma(x) = \frac{1}{1 + \exp(-x)} \quad (21)$$

then an Input gate, which includes two procedures, is used to update the information in cell state,

$$\begin{cases} i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \\ \tilde{C}_t = \tanh(W_C * [h_{t-1}, x_t] + b_C) \end{cases} \quad (22)$$

where  $i_t$  indicates the percentage of retained information in addable information while  $\tilde{C}_t$  is addable information. After obtaining the outputs from forgotten gate and input gate, a process is applied to combine them into cell state,

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (23)$$

Taking  $h_{t-1}$  as well as  $x_t$  as input through output gate in last step when the cell state has been updated,

$$\begin{cases} o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \\ h_t = o_t * \tanh(C_t) \end{cases} \quad (24)$$

**Table 2.** The composition and number of parameters of each layers in CNN. Reshape layer is utilized to change data format to desired format while the four following CNN layers are applied to exact the feature of sequence. At last, a full-connected layer change the feature map to solution.

Layer	Output Shape	Parameters
Reshape	(None, 1, 12)	0
Conv1D	(None, 1, 12)	156
Conv1D	(None, 1, 12)	312
Conv1D	(None, 1, 12)	600
Dropout	(None, 1, 12)	0
Conv1D	(None, 1, 12)	900
Dropout	(None, 1, 12)	0
Dense	(None, 1, 12)	259
Total Parameters: 2227		

where  $o_t$  presents the reserved part of hide cell state while  $h_t$  is the hide state of cell.

### 3. Experimental Method

As research background of this study, the principles and solving processes of four popular solutions are introduced in previous part. In this part, two neural network models would be established for resolving robot manipulator trajectory tracking. For trajectory tracking task, we want to obtain the transformation from transpose matrix  $T$  to each joints' angles  $\vec{\theta}$ . The relation could be as below,

$$(\theta_1, \theta_2, \dots, \theta_7) = \vec{\theta} \leftarrow {}^0_7T = \begin{pmatrix} a_{1,1} & a_{1,2} & a_{1,3} & x \\ a_{2,1} & a_{2,2} & a_{2,3} & y \\ a_{3,1} & a_{3,2} & a_{3,3} & z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (25)$$

where  $\vec{a}$  presents the rotation matrix while  $x, y, z$  indicates the location of manipulator end.

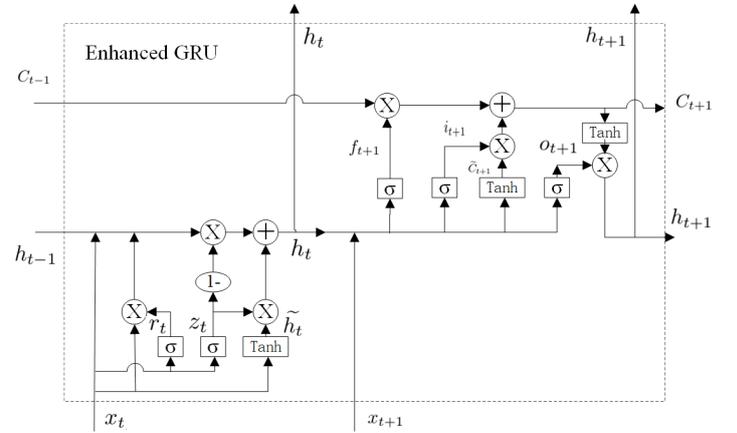
#### 3.1. CNN for Trajectory Tracking

A CNN architecture is proposed in this subsection to predict joint angles according to  ${}^0_7T$ , which contains six main layers shown in Fig. 2 (one reshape layer, four convolution layers as well as one dense layer).

As shown in (25), removing the bottom row without physical meaning of  ${}^0_7T$ , the rest part should be input data. A reshape layer is utilized to convert that  $3 \times 4$  matrix to 12-Dimensional vector. After that one convolution layer with 12 kernels will convolve the vector while next convolution layer with 24 kernels would convolve with the output. The convolution layer as below also has 24 kernels but with the operation 0.2 dropout. The standard CNN depends on gradient of each parameters to reduce otherness

**Table 3.** The architectures and features of enhanced GRU model. The whole model contain two enhanced GRU layers and two dropout layers to exact feature of sequence and improve generalization of model. Finally, a full-connected layer is linked at the end to transform cell state and hidden state to solution.

Layer	Output Shape	Parameters
Enhanced GRU	(None, 1, 12)	2136
Dropout	(None, 1, 12)	0
Enhanced GRU	(None, 1, 12)	2136
Dropout	(None, 1, 12)	0
Dense	(None, 1, 12)	91
Total Parameters: 4363		



**Figure 3.** The detail structure and component inside enhanced GRU.  $h_{t-1}$  and  $C_{t-1}$ , as the input of enhanced GRU, are hidden state and cell state of last cell. And these states will be added or subtracted information via the calculation of gate units, which takes two steps time  $t$ . At last,  $h_{t+1}$  as well as  $C_{t+1}$  is output as the result.

between prediction data and real data while it may cause over-fit due to the complex effect. The detail operation of dropout is applying  $p$  probability to drop one portion of neural cells and  $1 - p$  probability to reserve others. The deeper convolution layer with 36 kernels and 0.1 dropout operation. Finally, all the feature maps should go through the full-connected layer to extract feature from  $1 \times 36$  map to  $1 \times 7$  vector as the output, which are each joints' values.

#### 3.2. Enhanced GRU for Trajectory Tracking

The principles of GRU and cell state had been explained in section 2. It is a valid approach to improve model performance by appending cell state and gate unit into GRU[39–41]. Then a novel enhanced GRU is introduced here to acquire both excellent ability of convergence, real-time as well as wonderful performance.

The detail architecture and component of enhanced GRU is indicated in Fig. 3, including one cell state, one hidden cell state as well as five gate units. This architecture integrates reset gate, update gate, forgotten gate, input gate as well as output gate to assist cell state and hidden cell state to add or subtract information. And the unit takes two steps  $t$ , whose inputs are last unit's hidden state  $h_{t-1}$ , rare data  $x_t$  and  $x_{t+1}$  as well as last unit's cell state  $C_{t-1}$  while outputs are current unit's cell state  $C_{t+1}$  and hidden state  $h_{t+1}$ .

The derivation of the unit formula can be indicated as below. The expression of intermediate quantity  $h_t$  is as the same as that in (19). Merging (20) and (19), then,

$$\begin{aligned} f_{t+1} &= \sigma(W_f * [h_t, x_{t+1}] + b_f) \\ &= \sigma(W_f * [(1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, x_{t+1}] + b_f) \\ &= \sigma(W_f * [(1 - z_t) * h_{t-1} + z_t * \\ &\quad \tanh(W_{\tilde{h}} * [r_t * h_{t-1}, x_t]), x_{t+1}] + b_f) \end{aligned} \quad (26)$$

moreover, according to (22) and (19),  $i_{t+1}$  and  $\tilde{C}_{t+1}$  could be calculated by the following formula,

$$\begin{aligned} i_{t+1} &= \sigma(W_i * [h_t, x_{t+1}] + b_i) \\ &= \sigma(W_i * [(1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, x_{t+1}] + b_i) \\ &= \sigma(W_i * [(1 - z_t) * h_{t-1} + z_t * \tanh(W_{\tilde{h}} * \\ &\quad [r_t * h_{t-1}, x_t]), x_{t+1}] + b_i) \end{aligned} \quad (27)$$

$$\begin{aligned} \tilde{C}_{t+1} &= \tanh(W_C * [h_t, x_{t+1}] + b_C) \\ &= \tanh(W_C * [(1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, x_{t+1}] + b_C) \\ &= \tanh(W_C * [(1 - z_t) * h_{t-1} + z_t * \tanh(W_{\tilde{h}} * \\ &\quad [r_t * h_{t-1}, x_t]), x_{t+1}] + b_C) \end{aligned} \quad (28)$$

whereupon  $C_{t+1}$  can be presented as below,

$$C_{t+1} = f_{t+1} * C_{t-1} + i_{t+1} * \tilde{C}_{t+1} \quad (29)$$

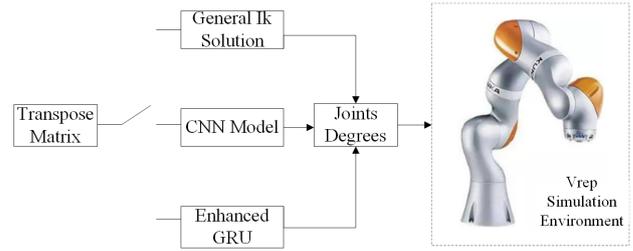
then with (24) and (19),

$$\begin{aligned} o_{t+1} &= \sigma(W_o * [h_t, x_{t+1}] + b_o) \\ &= \sigma(W_o * [(1 - z_t) * h_{t-1} + z_t * \tilde{h}_t, x_{t+1}] + b_o) \\ &= \sigma(W_o * [(1 - z_t) * h_{t-1} + z_t * \tanh(W_{\tilde{h}} * \\ &\quad [r_t * h_{t-1}, x_t]), x_{t+1}] + b_o) \end{aligned} \quad (30)$$

At last,

$$h_{t+1} = o_{t+1} * \tanh(C_{t+1}) \quad (31)$$

So far, the solution procedure of enhanced GRU has finished. To meet the enhanced GRU's solution modality,  ${}^0_7T$  is flattened to 12-Dimensional tensor. And in enhanced GRU model, two cascade enhanced GRU with dropout layers are utilized to extract sequence feature from the input tensor as shown in Fig. 3. At last, 7 full-connected neural cell is applied to transform the feature state to each joints' angles  $\vec{\theta}$ .



**Figure 4.** The interaction methodology with Vrep simulation environment. As a input, transpose matrix could be sent to several inverse kinematics models. After obtaining each joints values, controller would transmit these data to vrep simulation environment via python remote client. Finally, the visual simulation is presented in Vrep.

## 4. Applications, Comparisons and Tests

Performance analysis by using KUKA iiwa LBR 7 as experimental robot manipulator in V-rep simulation environment via three different trajectory-tracking tasks to indicate the feasibility of enhanced GRU model. As the interaction approach with coding environment and V-rep simulation environment shown in Fig. 4, the peripheral controller would obtain each joints' angles  $\vec{\theta}$  while three varieties of methods are utilized to resolve the transpose matrix. After that,  $\vec{\theta}$  would be sent to V-rep simulation environment through python remote client while the location of manipulator end would be returned. After the comparisons on accuracy performance in space distance with other inverse kinematics solutions in the case are record. Due to the discontinuity of general inverse kinematics, it did not complete the test and is not utilized as a comparison object in section 4.2 while the real performance of it will be presented at the end of this section.

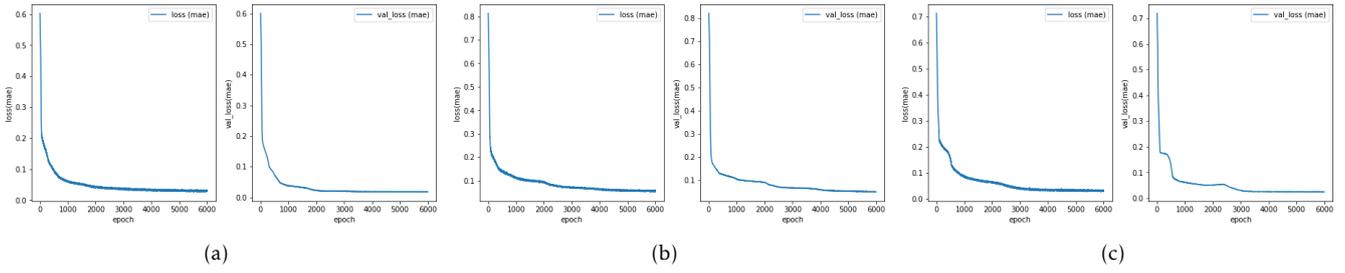
### 4.1. Enhanced GRU Training Process

Tensorflow 2.0 is applied in this paper to build enhanced GRU model as Fig. 3. Before training, the input data should be standardized to improve the generalization of deep learning model, and the standard method is as below,

$$\tilde{x} = \frac{x - \bar{x}}{x_{max} - x_{min}} \quad (32)$$

where  $\tilde{x}$  is the standardized data while  $x$  presents original data,  $\bar{x}$  means the average value of  $x$ ,  $x_{max}$  and  $x_{min}$  indicates the max and min value of  $x$  respectively.

After that, standardized data  $\tilde{x}$  is split to training dataset and validation dataset with the proportion four to one, which provides the performance on un-trained dataset and meanwhile enhances the generalization ability of model. Through the training process, mean absolute error (mae) is utilized to analyse the quality



**Figure 5.** The loss value (mae) through the whole training process with enhanced GRU model (total 6000 epochs with Adam algorithm). And due to the higher difficulty of trajectory 2, the loss in trajectory 2 through the whole training process would be larger than others. (a) training in trajectory 1. (b) training in trajectory 2. (c) training in trajectory 3.

of model and the calculation formula is as below,

$$loss = \frac{1}{nm} * \sum_{i=0}^m \sum_{j=0}^n (|\tilde{\theta}_{i,j} - \theta_{i,j}|) \quad (33)$$

the  $n$  in (33) is the number of samples in one batch,  $m$  means the number of joints, the  $\tilde{\theta}_{i,j}$  presents the predict  $j$ th joint value in  $i$ th sample while  $\theta_{i,j}$  indicates the real  $j$ th joint angle in  $i$ th sample[42, 43].

All the three trajectories are made up of 450 points and presented as sub-picture (a) in Fig. 6, Fig.7 as well as Fig. 8. Trajectory 1 has the largest spans in Z-axis and small inflection point while the main trajectory of trajectory 2 is like s-shape with few span in Z-axis and contain various of inflection points. From the top view, trajectory 3 is similar with M-shape and the latter half of it is straight line motion.

The whole training process contains 6000 epochs with loss optimized algorithm Adam and 128 batch size. The change maps of loss value through the whole training process are illustrated in Fig. 5 and (a),(b) as well as (c) present the train process from trajectory 1 to 3 respectively. As (a) in Fig. 5 indicated, the loss value of training dataset reached slightly less than 0.1 on 1000 epochs while it settled the number nearby 0.05 on about 5000 epochs. The situation is similar in validation dataset of trajectory 1 but with lower loss value. Due to the higher trajectory difficulty of trajectory 2, the loss values of both datasets are slightly over than 0.1 on 1000 epochs while the values decline to approximately 0.07 on 6000 epochs. The loss map is a bit zigzag in last training of trajectory 2, which attain probably 0.1 on neighbouring 1000 epochs while it is cut a half on 3000 epochs.

#### 4.2. Comparison with Standard GRU and CNN Model

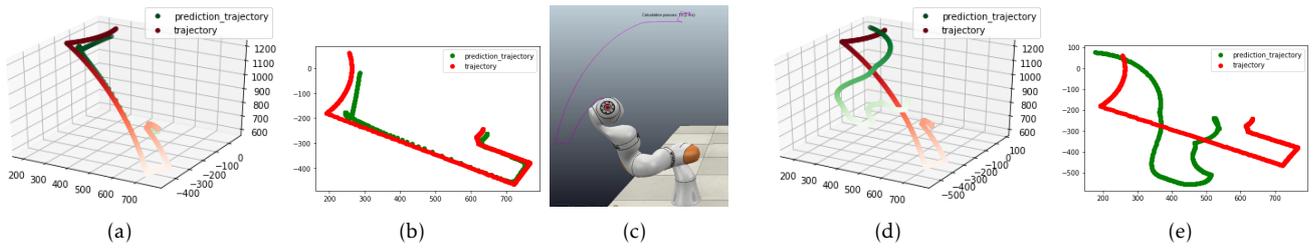
In order to investigate the otherness between enhanced GRU, standard GRU and CNN model, the same trajectory tracking tasks have tested by both of them (the two

**Table 4.** Space distance via enhanced GRU Model (31), standard GRU model as well as CNN model between prediction trajectory and desired trajectory among the three trajectories tracking tasks

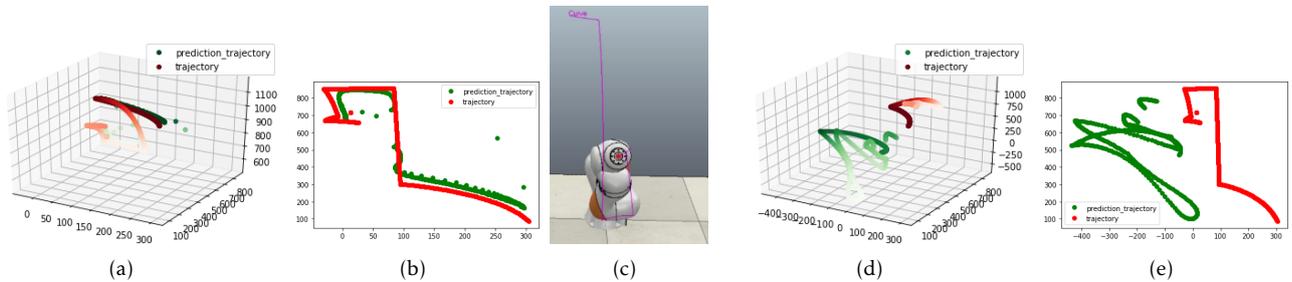
Model	Trajectory 1	Trajectory 2	Trajectory 3
Enhanced GRU Model (mm)	26.52	91.07	33.28
Standard GRU Model (mm)	40.44	116.35	46.20
CNN Model (mm)	219.97	1130.00	378.80

other neural network models' training processes copied from enhanced GRU's). The theoretical performances of them are calculated by forward kinematics and indicated in Fig. 6 to Fig. 8. The trajectories generated by standard GRU model are not indicated in these picture on account of the visual differences between them with enhanced GRU's, but the quantitative analysis (mean space distance) of the discrepancy between them is illustrated in Table 4.

In 3D view of the sub-figure (a) in Fig. 6, the real trajectory and prediction trajectory are almost overlap while the prediction trajectory is a bit offset at the corner on the top. The situation becomes clearer in top view of the sub-figure (b) in Fig. 6 that real trajectory has a little crooked while the prediction trajectory slides diagonally down from the top at the corner on the left. After first corner, the gap between the two trajectories narrows. The prediction trajectory is also similar to real trajectory while some points are out of desired trajectory in 3d view of the sub-figure (a) in Fig. 7. While the horizon changes to top, the sawtooth of prediction trajectory becomes more distinct but the general trajectories are resemblance. It should be pointed out that prediction trajectory has distinct radian when it comes to a corner. Some scattered points of prediction trajectory are particularly evident in 3D



**Figure 6.** Motion results of trajectory 1 via enhanced GRU model (31) and CNN model with a robot manipulator to track trajectory 1 (The trajectories generated by standard GRU model are not indicated in these pictures on account of the visual differences between them with enhanced GRU's, but the quantitative analysis (mean space distance) of the discrepancy between them is illustrated in Table 4). The result created by general IK solution is not presented because the multiple solutions as well as singular of general IK solution causes the stuck of Vrep. The trajectory generated by enhanced GRU model performs better while at first corner there is still a slight difference with desired trajectory. In contrast, the prediction trajectory of CNN presents that the CNN model is under-fitting (a) Motion process in 3D view (enhanced GRU). (b) Motion process in top view (enhanced GRU). (c) The desired trajectory in Vrep. (d) Motion process in 3D view (CNN). (e) Motion process in top view (CNN).



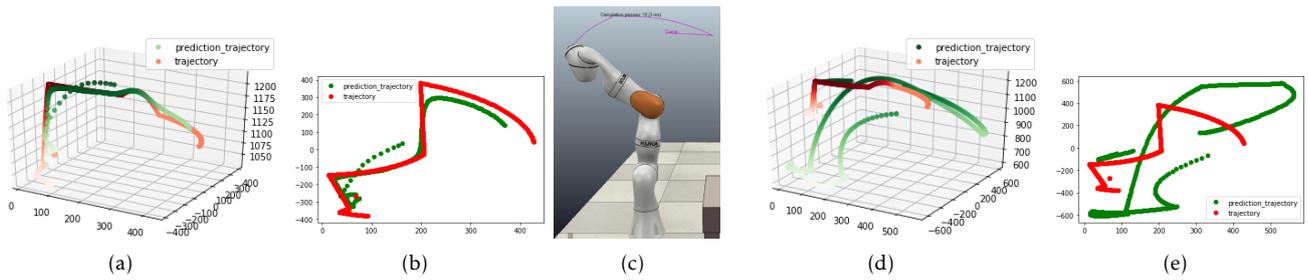
**Figure 7.** Motion results of trajectory 1 via enhanced GRU model (31) and CNN model with a robot manipulator to track trajectory 2 (The trajectories generated by standard GRU model are not indicated in these picture on account of the visual differences between them with enhanced GRU's, but the quantitative analysis (mean space distance) of the discrepancy between them is illustrated in Table 4). The result created by general IK solution is not presented because the multiple solutions as well as singular of general IK solution causes the stuck of Vrep. The trajectory generated by enhanced GRU model performs better while also at corners there still be some slight differences with desired trajectory. In contrast, the prediction trajectory of CNN presents that the CNN model is not convergence. (a) Motion process in 3D view (enhanced GRU). (b) Motion process in top view (enhanced GRU). (c) The desired trajectory in Vrep. (d) Motion process in 3D view (CNN). (e) Motion process in top view (CNN).

view of the sub-figure (a) in Fig. 8, which drops from the top. Prediction trajectory goes around in circles at the lower left corner while it performs well at the rest part in top view.

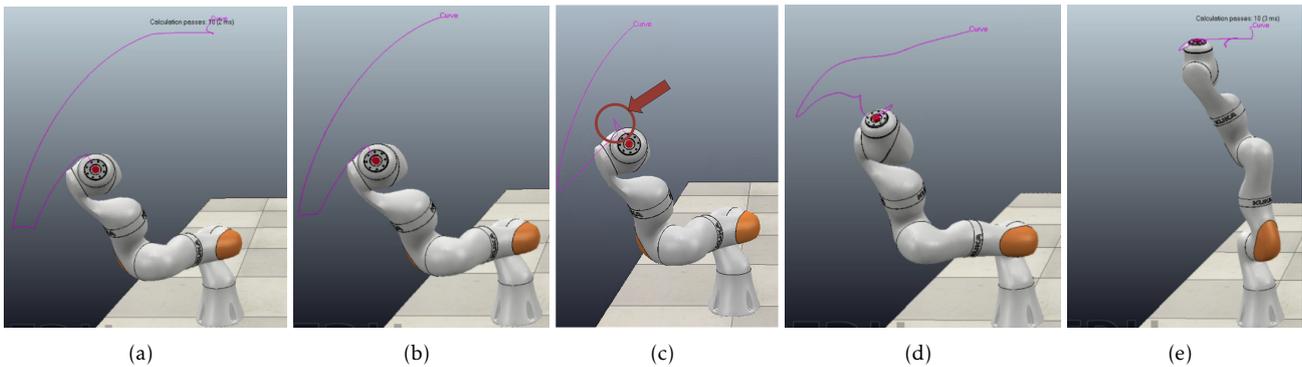
The condition is bad in the case of CNN model. Visible differences can be witnessed in sub-figure (c) of Fig. 6, by contrast, the prediction trajectory's shape is a little similar with desired trajectory although the gap between them in sub-figure (d) of Fig. 6 is always large during the tracking task. The same situation happens in the case of sub-figure (d) of Fig. 8. The shape of prediction trajectory is a little bit resemble real trajectory in the top of view while they are strangers in 3D view. Things change in sub-figure (c) and (d) in Fig. 7 that the prediction trajectory and desired trajectory become two separate curves. The gap between them are visible huge both in 3D view or from the top of view.

For a better difference analysis between prediction trajectory and real trajectory, average Cartesian space distance is calculated and presented in table 4. The largest space distance of both of the two models is in trajectory 2 tracking task, which means the difficulty is the greatest. The average space distances between prediction trajectory and desired trajectory of enhanced GRU model as well as CNN model are 181.56mm and 1130.00mm respectively. The smallest space distance is indicated in trajectory 1 tracking task and the values of them are 126.85mm and 219.97mm severally. What should be pointed out is that all the trajectories tracking tasks via enhanced GRU model are below 200mm, while presents the feasibility of enhanced GRU model in trajectory tracking task.

To intuitively investigate the trajectories, this paper applied these models in the case in trajectory tracking



**Figure 8.** Motion results of trajectory 1 via enhanced GRU model (31) and CNN model with a robot manipulator to track trajectory 3 (The trajectories generated by standard GRU model are not indicated in these picture on account of the visual differences between them with enhanced GRU’s, but the quantitative analysis (mean space distance) of the discrepancy between them is illustrated in Table 4). The result created by general IK solution is not presented because the multiple solutions as well as singular of general IK solution causes the stuck of Vrep. The gap between desired trajectory and prediction trajectory by enhanced GRU is less. Although the difference is very less, one point should be point out is that enhanced GRU performs better in arc angle. The CNN model requires more than 6000 epochs to converge on account of the bad performance in trajectory 3 tracking task. (a) Motion process in 3D view (enhanced GRU). (b) Motion process in top view (enhanced GRU). (c) The desired trajectory in Vrep. (d) Motion process in 3D view (CNN). (e) Motion process in top view (CNN).



**Figure 9.** The performance of each models in Vrep to track trajectory 1. The results are consistent with the previous theoretical analysis, whatever the trajectory or pose of robot controlled by enhanced GRU is most similar with desired consequence. The pose and trajectory created by standard GRU is similar with enhanced GRU’s, but it shows less precise in detail pointed in sub-figure (c). The result generated by CNN model would be better if training CNN model more epochs. Due to the multiple solution and singular of general IK solution, Vrep cannot react immediately then it causes stuck. (a) Desired Trajectory. (b) Enhanced GRU. (c) Standard GRU. (d) CNN. (e) General IK Solution.

task to be visualized and simulated in Vrep. As it is illustrated in Fig. 9, the best tracking performance among the three solutions belongs to enhanced GRU while general IK solution reached the worst consequence. Because the solution generated by general IK solution is not related to the solution in last time  $t - 1$ , the pose would change a bit, which cause the stuck of Vrep, although the trajectory was correct. The manipulator controlled by CNN has the resemble pose with desired motion but with low accuracy to real trajectory. There is only one comparison in Vrep indicated in this section, and you can have a full observation of performance in the enclosure video.

To be summarize, general IK solution has excellent ability with solving a single problem but weak to deal

with continuous issue. In addition, due to the multiple-solutions as well as singular of general IK solution, it could not be a wonderful choice to resolve a trajectory tracking. The standard GRU can converges promptly while lacks less precise in detail, especially at corners. The potential of CNN was not fully developed in this section on account of its long convergence time. The enhanced GRU model indicated the best performance among the three solutions whatever in training process, otherness of trajectories and visual simulation.

### 5. Conclusion

In conclusion, this paper introduces a novel enhanced GRU to solve the trajectory tracking of robot manipulator. By incorporating GRU model with cell state as

well as gate units, the enhanced GRU model achieve the target of low convergence as well as high accuracy of trajectory tracking and pose imitation. The main principle of this study is to unite the mixed hidden state as well as the cell state to build one hybrid unit to solve the slightly low performance problem in GRU. The derivations about the calculation of cell state and mixed hidden state of presented model has been indicated. At last, three trajectories tracking tasks, comparison and visual simulation have been testified the feasibility as well as superiority of enhanced GRU model.

## References

- [1] W. Si, G. Srivastava, Y. Zhang, L. Jiang, "Green Internet of Things Application of a Medical Massage Robot With System Interruption," *IEEE Access*, vol. 7, pp. 127066-127077, 2019.
- [2] H. Wang, Y. Huang, S. Xiao, Y. Lin, Z. Jiang, "Design of A New Parallel Metamorphic Robot for Chinese Medicine Massage," *Basic and Clinical Pharmacology and Toxicology*, vol. 124, no. 3, pp. 39-40, Apr. 2019.
- [3] Z. Pang, B. Zhang, J. Yu, Z. Sun, L. Gong, "Design and Analysis of a Chinese Medicine Based Humanoid Robotic Arm Massage System," *Applied Science-Basel*, vol. 9, no. 20, Oct. 2019.
- [4] S. Kumar, K. Madhava, S. Srikanth, "Planning non-holonomic stable trajectories on uneven terrain through non-linear time scaling," *Mechanism and Machine Theory*, vol. 40, no. 8, pp. 1419-1440, Dec. 2016.
- [5] O. Kayumov, "Planning the Subminimum-Time Motion of a Robotic Manipulator Which Avoids Obstacles," *Journal of Computer and System International*, vol. 55, no. 6, pp. 865-877, Nov. 2016.
- [6] K. Dumlu, "Trajectory Tracking Control for a 3-DOF Parallel Manipulator Using Fractional-Order Control," *IEEE Trans. Ind. Electron.*, vol. 61, no. 7, pp. 3417-3426, Aug. 2013.
- [7] M. Galicki, "Finite-time trajectory tracking control in a task space of robotic manipulators," *Automatica*, vol. 67, pp. 165-170, May. 2016.
- [8] A. Duka, "Neural network based inverse kinematics solution for trajectory tracking of a robotic arm," *Procedia Technology*, vol. 12, pp. 20-27, Dec. 2013.
- [9] H. Peng, F. Li, J. Liu, Z. Ju, "A Symplectic Instantaneous Optimal Control for Robot Trajectory Tracking With Differential-Algebraic Equation Models," *IEEE Trans. Ind. Electron.*, vol. 67, no. 5, pp. 3819-3829, May. 2020.
- [10] L. Truong, S. Huang, V. Yen, P. Cuong, "Adaptive Trajectory Neural Network Tracking Control for Industrial Robot Manipulators with Deadzone Robust Compensator," *International Journal of Control Automation and System*, [online], Doi: 10.1007/s12555-019-0513-7.
- [11] I. Komnik, M. Peters, J. Funken, S. David, S. Weiss, W. Potthast, "Non-Sagittal Knee Joint Kinematics and Kinetics during Gait on Level and Sloped Grounds with Unicompartmental and Total Knee Arthroplasty Patients," *PLoS One*, vol. 11, no. 12, Dec. 2016.
- [12] Y. Zhao, S. Hu, Y. Yang, "Inverse kinematics of asymmetric octahedral variable geometry truss manipulator with obstacle avoidance through inexact interior point optimization," *International Journal of Advanced Robotic System*, vol. 15, no. 6, Dec. 2018.
- [13] J. Kim, J. Lee, Y. Choi, "PiROB: Vision-based pipe-climbing robot for spray-pipe inspection in nuclear plants," *International Journal of Advanced Robotic System*, vol. 15, no. 6, Dec. 2018.
- [14] Z. Zake, F. Chaumette, N. Pedemonte, S. Caro, "Robust 21/2D Visual Servoing of A Cable-Driven Parallel Robot Thanks to Trajectory Tracking," *IEEE Robotics and Automation Letter*, vol. 5, no. 2, pp. 660-667, Apr. 2020.
- [15] M. wang, Z. Lu, H. Li, W. Jiang, Q. Liu, "genCNN: A Convolutional Architecture for Word Sequence Prediction," *ACL 2015*, arXiv: 1503.05034.
- [16] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, Nov. 1997.
- [17] M. Gao, G. Shi, S. Li, "Online Prediction of Ship Behavior with Automatic Identification System Sensor Data Using Bidirectional Long Short-Term Memory Recurrent Neural Network," *Sensors*, vol. 18, no. 12, Dec. 2018.
- [18] P. Shrey, K. Thenkurussi, "IART: Learning From Demonstration for Assisted Robotic Therapy Using LSTM," *IEEE Robotics and Automation Letter*, vol. 5, no. 2, pp. 477-484, Apr. 2020.
- [19] D. Robert, A. Narges, M. Anand, W. Madeleine, L. GI, L. MR, V. SS, H. GD, "Segmenting and classifying activities in robot-assisted surgery with recurrent neural networks," *International Journal of Computer assisted Radiology and Surgery*, vol. 14, no. 11, pp. 2005-2020, Nov. 2019.
- [20] T. Shibahara, K. Yamanishi, et al. "Event De-Noising Convolutional Neural Network for Detecting Malicious URL Sequences from Proxy Logs," *IEICE Trans. Fundamentals of Electronic Communication and Computer Science*, vol. E101A, no. 12, pp. 2149-2161, Dec. 2018.
- [21] Y. Li, S. Li, D. Caballero, M. Miyasaka, A. Lewis and B. Hannaford, "Improving control precision and motion adaptiveness for surgical robot with recurrent neural network," *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vancouver, BC, 2017, pp. 3538-3543.
- [22] S. Li, H. Wang, M. Rafique, "A Novel Recurrent Neural Network for Manipulator Control Improved Noise Tolerance," *IEEE Trans. Neural Netw. Syst.*, vol. 29, no. 5, pp. 1908-1918, Apr. 2017.
- [23] J. Wu, J. She, Y. Wang, C. Su, "Position and Posture Control of Planar Four-Link Underactuated Manipulator Based on Neural Network Model," *IEEE Trans. Ind. Electron.*, vol. 67, no. 6, pp. 4721-4728, Jun. 2020.
- [24] S. Li, Y. Zhang, L. Jin, "Kinematic Control of Redundant Manipulators Using Neural Networks," *IEEE Trans. Neural Network and Learning Systems*, vol. 28, no. 10, pp. 2243-2254, Oct. 2017.
- [25] Z. Zhang, L. Zheng, Z. Chen, L. Kong, H. Reza, "Mutual-Collision-Avoidance Scheme Synthesized by Neural Networks for Dual Redundant Robot Manipulators Executing Cooperative Tasks," *IEEE Trans. Neural Network and Learning Systems*, Apr. 2020, Doi: 10.1109/TNNLS.2020.2980038.

- [26] H. Jiang, G. Xu, W. Zeng, F. Gao, "Design and kinematic modeling of a passively-actively transformable mobile robot," *Mechanism and Machine Theory*, vol. 142, Dec. 2019.
- [27] Y. Chen, X. Luo, B. Han, Y. Jia, G. Liang, X. Wang, "A General Approach Based on Newton's Method and Cyclic Coordinate Descent Method for Solving the Inverse Kinematics," *Applied Science-Basel*, vol. 9, no. 24, Dec. 2019.
- [28] Y. Wan, Z. Han, J. Zhong, G. Chen, "Pattern recognition and bionic manipulator driving by surface electromyography signals using convolutional neural network," *International Journal of Advanced Robotic Systems*, vol. 15, no. 5, Oct. 2018.
- [29] C. Yang, F. Yao, M. Zhang, "Adaptive Backstepping Terminal Sliding Mode Control Method Based on Recurrent Neural Networks for Autonomous Underwater Vehicle," *Chinese Journal of Mechanical Engineering*, vol. 31, no. 1, Dec. 2018.
- [30] M. Er, Y. Zhang, N. Wang, M. Pratama, "Attention pooling-based convolutional neural network for sentence modelling," *Information Science*, vol. 373, pp. 388-403, Dec. 2016.
- [31] X. Liu, Z. Pan, H. Yang, X. Zhou, W. Bai, X. Niu, "An Adaptive Moment estimation method for online AUC maximization," *PLoS One*, vol. 14, no. 4, Apr. 2019.
- [32] E. Tsironi, P. Barros, C. Weber, S. Wermter, "An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition," *Neurocomputing*, vol. 268, pp. 76-86, Dec. 2017.
- [33] R. Zhao, D. Wang, R. Yan, K. Mao, F. Shen, J. Wang, "Machine Health Monitoring Using Local Feature-Based Gated Recurrent Unit Networks," *IEEE Trans. Ind. Electron.*, vol. 65, no. 2, pp. 1539-1548, Feb. 2018.
- [34] S. Li, J. He, U. Rafique, and Y. Li, "Distributed Recurrent Neural Networks for Cooperative Control of Manipulators: A Game-Theoretic Perspective", *IEEE Transactions on Neural Networks and Learning Systems*, 2017, 28(2): 415 - 426.
- [35] S. Li, Y. Zhang and L. Jin, "Kinematic Control of Redundant Manipulators Using Neural Networks", *IEEE Transactions on Neural Networks and Learning Systems*, 2017, 28(10): 2243-2254.
- [36] S. Li, H. Wang and U. Rafique, "A Novel Recurrent Neural Network for Manipulator Control with Improved Noise Tolerance", *IEEE Transactions on Neural Networks and Learning Systems*, 2018, 29.5: 1908-1918.
- [37] S. Li, Y. Guo, "Distributed source seeking by cooperative robots: All-to-all and limited communications", *ICRA*, 2012, DOI: 10.1109/ICRA.2012.6224713
- [38] X. Jiang, S. Li, "BAS: Beetle Antennae Search Algorithm for Optimization Problems", *International Journal of Robotics and Control*, vol.1, no.1, 2018
- [39] AH. Khan, S. Li, D. Chen, "Tracking control of redundant mobile manipulator: An RNN based metaheuristic approach", *Neurocomputing*, 2020, 400(4), 272-284.
- [40] Z. Li, S. Li, "Neural Network Model-Based Control for Manipulator: An Autoencoder Perspective", *IEEE Transactions on Neural Networks and Learning Systems*, 2021, 14.9: 1-15.
- [41] D. Wu, Y. He, MC. Zhou, "A latent factor analysis-based approach to online sparse streaming feature selection", *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2021, 4.8: 1-15.
- [42] M. Shang, Y. Yuan, X. Luo, "An  $\alpha$ - $\beta$  -Divergence-Generalized Recommender for Highly Accurate Predictions of Missing User Preferences", *IEEE Transactions on Cybernetics*, 2021, 18.1: 1-13.
- [43] Z. Li, S. Li, X. Luo, "An Overview of Calibration Technology of Industrial Robots", *IEEE/CAA Journal of Automatica Sinica*, 2021, Volume: 8, Issue: 1, Pages: 23 - 36.