A Visual-Based Gesture Prediction Framework Applied in Social Robots

Bixiao Wu, Junpei Zhong, *Senior Member, IEEE*, and Chenguang Yang, *Senior Member, IEEE*

***Abstract*—In daily life, people use their hands in various ways for most daily activities. There are many applications based on the position, direction, and joints of the hand, including gesture recognition, gesture prediction, robotics and so on. This paper proposes a gesture prediction system that uses hand joint coordinate features collected by the Leap Motion to predict dynamic hand gestures. The model is applied to the NAO robot to verify the effectiveness of the proposed method. First of all, in order to reduce jitter or jump generated in the process of data acquisition by the Leap Motion, the Kalman filter is applied to the original data. Then some new feature descriptors are introduced. The length feature, angle feature and angular velocity feature are extracted from the filtered data. These features are fed into the long-short time memory recurrent neural network (LSTM-RNN) with different combinations. Experimental results show that the combination of coordinate, length and angle features achieves the highest accuracy of 99.31%, and it can also run in real time. Finally, the trained model is applied to the NAO robot to play the finger-guessing game. Based on the predicted gesture, the NAO robot can respond in advance.**

***Index Terms*—Finger-guessing game, gesture prediction, human-robot interaction, long-short time memory recurrent neural network (LSTM-RNN), social robot.**

I. Introduction

CURRENTLY, computers are becoming more and more popular, and the demand for human-robot interaction is increasing. People pay more attention to research of new technologies and methods applied to human-robot interactions [1]–[3]. Making human-robot interaction as natural as daily human-human interaction is the ultimate goal. Gestures have always been considered an interactive technology that can provide computers with more natural, creative and intuitive methods. Gestures have different meanings in different disciplines. In terms of interaction design, the difference between using gestures and using a mouse and keyboard, etc., is obvious, i.e., gestures are more acceptable to people. Gestures are comfortable and less limited by interactive devices, and they can provide more information. Compared with traditional keyboard and mouse control methods, the direct control of the computer by hand movement has the advantages of being natural and intuitive.

Gesture recognition [4] refers to the process of recognizing the representation of dynamic or static gestures and translating them into some meaningful instructions. It is an extremely significant research direction in the area of human-robot interaction technology. The method of realizing gesture recognition can be divided into two types: visual-based [5], [6] gesture recognition and non-visual-based gesture recognition. The study of non-vision approaches began in the 1970s. Non-vision methods always take advantage of wearable devices [7] to track or estimate the orientation and position of fingers and hands. Gloves are very common devices in this field, and they contain the sensory modules with a wired interface. The advantage of gloves is that their data do not need to be preprocessed. Nevertheless, they are very expensive for virtual reality applications. They also have wires, which makes them uncomfortable to wear. With the development of technology, current research on non-visual gesture recognition is mainly focused on EMG signals [8]–[11]. However, EMG signals are greatly affected by noise, which makes it is difficult to process.

Gesture recognition is based on vision and is less intrusive and contributes to a more natural interaction. It refers to the use of cameras [12]–[16], such as Kinect [17], [18] and Leap Motion [19], [20], to capture images of gestures. Then some algorithms are used to analyze and process the acquired data to get gesture information, so that the gesture can be recognized. It is also more natural and easy to use, becoming the mainstream way of gesture recognition. However, it is also a very challenging problem.

By using the results of gesture recognition, the subsequent gesture of performers can be predicted. This process could be called gesture prediction, and it has wider applications. In recent years, with the advent of deep learning, many deep neural networks (DNN) are applied to gesture prediction. Zhang *et al*. [21] used an RNN model to predict gestures from raw sEMG signals. Wei *et al*. [22] combined a 3D convolutional residual network and bidirectional LSTM network to recognize dynamic gesture. Kumar *et al*. [23] proposed a multimodal framework based on hand features captured from Kinect and Leap Motion sensors to recognize gestures, using a

Manuscript received April 18, 2021; revised May 23, 2021 and June 5, 2021; accepted June 22, 2021. This work was supported in part by National Nature Science Foundation of China (NSFC) (U20A20200, 61861136009), in part by Guangdong Basic and Applied Basic Research Foundation (2019B1515120076, 2020B1515120054), in part by Industrial Key Technologies R & D Program of Foshan (2020001006308). Recommended by Associate Editor Hui Yu. *(Corresponding author: Chenguang Yang.)*

Citation: B. X. Wu, J. P. Zhong, and C. G. Yang, "A visual-based gesture prediction framework applied in social robots," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 3, pp. 510–519, Mar. 2022.

B. X. Wu and C. G. Yang are with the College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China (e-mail: wubixiao1997@163.com; cyang@ieee.org).

J. P. Zhong is with the Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 511442, China (e-mail: jonizhong@scut.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2021.1004243

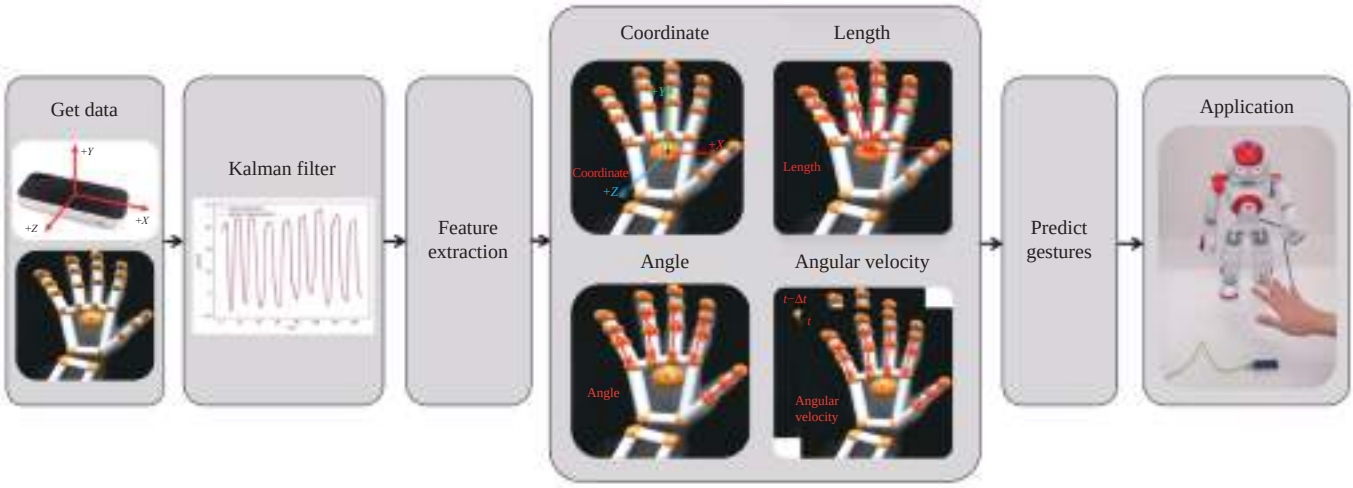


Fig. 1. Pipeline of the proposed approach.

hidden Markov model (HMM) and bidirectional long short-term memory model (LSTM). The LSTM [24] has become an effective model for solving some learning problems related to sequence data. Hence, inspired by the previous works, we adopt the LSTM to predict gestures in our proposed framework.

In the method of gesture prediction, hand key point detection is one of the most important steps. In the early stage of technological development, the former mainly used color filters to segment the hands to achieve detection. However, this type of method relies on skin color, and the detection performance is poor when the hand is in a complex scene. Therefore, the researchers proposed a detection method based on 3D hand key points. The task goal of the 3D hand key point estimation is to locate the 3D coordinates of hand joints in a frame of depth image, mostly used in virtual immersive games, interactive tasks [25], [26], and so on. Leap Motion is a kind of equipment for 3D data extraction based on vision technology. This device could extract the position of the hand joints, orientation and the speed of the fingertips movement. Recently, Leap Motion has always been used by researchers for gesture recognition and prediction. For example, some scholars use it to recognize American sign language (ASL) [27], [28], and it has a high gesture recognition accuracy. Moreover, Zeng *et al*. [29] proposed a gesture recognition method based on deterministic learning and joint calibration of the Leap Motion. And Marin *et al*. [30] developed a method to combine the Leap Motion and Kinect to calculate different features of hand, and a higher accuracy was obtained. In this work, we use the data of hand key points detected by the Leap Motion to predict gestures and utilize the gesture recognition results to play the finger-guessing game. This game contains three gestures: rock, paper and scissors. The winning rules of this game are: scissors wins paper, paper wins rock, rock wins scissors. Based on these game rules, this paper proposes a method to judge gestures in advance when the player has not completed the action.

The combination of the Leap Motion and LSTM significantly improves human-robot interaction. The Leap Motion could track each joint of the hand directly and has the ability to recognize or predict gestures. Moreover, compared with other devices, the Leap Motion has higher localization precision. On the other hand, the LSTM can solve the prediction problem well in most cases, and it is one of the important algorithms of deep learning (DL). This work combines the strengths of the LSTM and Leap Motion to predict gestures. Leap Motion captures 21 three-dimensional joint coordinates in each frame, and the LSTM network is used to train and test these features. This work has some novel contributions:

1) A method for predicting gestures based on the LSTM is proposed. The data of gestures is collected by the Leap Motion.

2) In order to reduce or eliminate the jitter or jump generated in the process of acquiring data by the Leap Motion, the Kalman filter is applied to solve this problem effectively.

3) We propose a reliable feature extraction method, which extracts coordinate features, length features, angle features and angular velocity features, and combines these features to predict gestures.

4) We apply the trained model to the NAO robot and make it play the finger-guessing game with players, which effectively verifies the real-time and accuracy of the proposed approach.

The rest part of this paper is structured as below: in Section II, the process of processing data is given. In Section III, the experiment of this work is introduced in detail and the effectiveness is verified in this section. Finally, Section IV makes a summary. The framework of this paper is shown in Fig. 1.

II. Data Processing

A. Leap Motion Controller

The structure of the Leap Motion is not complicated, as shown in Fig. 2. The main part of the device includes two cameras and three infrared LEDs. They tracked infrared light outside the visible spectrum, which has a wavelength of 850 nanometers. Compared with other depth cameras, such as the Kinect, the information obtained from the Leap Motion is

Fig. 2. The structure of the Leap Motion.

limited (only a few key points rather than complete depth information) and it works in smaller three-dimensional areas. However, it is more accurate to use Leap Motion to acquire data. Moreover, Leap Motion provides software that can recognize some movement patterns, including swipe, tap and so on. Developers can access some functions of Leap Motion through the application programming interface (API) to create new applications. For example, they can obtain information about the position and length of the user's hand to recognize different gestures.

Even though the manufacturers declare an accuracy of the Leap Motion in position measurement is around 0.01 mm, [31] shows that it is about 0.2 mm for static measurements and 0.4 mm for dynamic measurements in fact. And in the coordinates of the finger joints extracted by Leap Motion, there exists jitter or even jump, which could affect the accuracy of the experimental results. In order to reduce or eliminate these phenomena, this work takes advantage of the Kalman filter to correct the predicted position of hand joints.

B. Data Acquisition

Each finger is marked with a name: thumb, index, middle, ring, and pinky, including four bones (except thumb). As shown in Fig. 3, the phalanx of the finger includes the metacarpal, proximal phalanx, middle phalanx, and distal phalanx. Particularly, the thumb has only three phalanges, one less than the other. In the algorithm design, we set the length of the thumb metacarpal bone to 0 to guarantee that all five fingers have the same number of phalanges, which is easy to programme. In this work, the main data acquired by the Leap Motion are as follows:

1) Number of Detected Fingers: $Num \in [1,5]$ is the number of fingers detected by Leap Motion.

2) Position of the Finger Joints: $P_i, i = 1,2,3,\ldots,20$ contains the three-dimensional position of each finger joint. The Leap Motion provides a one-to-one map between coordinates and finger joints.

3) Palm Center: $P_c(x_0, y_0, z_0)$ represents three-dimensional coordinates of the center of the palm area in 3D space.

4) Fingertips Movement Speed: V represents the speed in the three-dimensional direction of each fingertip detected by the Leap Motion.

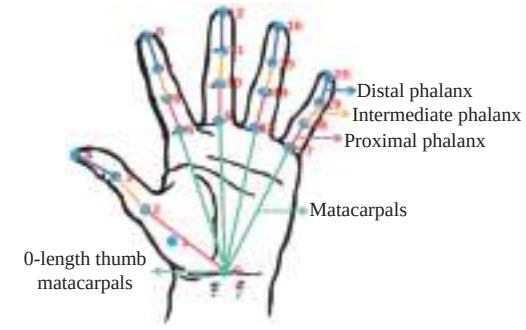


Fig. 3. Definition of endoskeleton in Leap Motion.

C. Kalman Filter

1) Problem Formulation: In the process of gesture changes, the fingertips have the largest range of change and can more easily jitter or jump than other joints, therefore, the Kalman filter is used to process the data from fingertips. Compared with other filters, such as the particle filter, the Luenburger observer filter, etc., the Kalman filter has sufficient accuracy and can effectively remove Gaussian noise. In addition, its low computational complexity meets the real-time requirements of this work. Therefore, the Kalman filter is used for this work.

Suppose that the current position of the fingertips obtained by Leap Motion is P_t, and the speed is V_t. The Kalman filter assumes that these two variables obey a Gaussian distribution, and each variable has a mean value of μ, and variance of σ^2. For clarity, X_t denotes the best estimate at time t, and Y_t denotes the covariance matrix. The equations of X_t and Y_t are as follows:

$$X_t = \begin{bmatrix} P_t \\ V_t \end{bmatrix} \tag{1}$$

$$Y_t = \begin{bmatrix} \sum_{PP} & \sum_{PV} \\ \sum_{VP} & \sum_{VV} \end{bmatrix}. \tag{2}$$

2) The Prediction Process: We need to predict the current state (time t) according to the state of the last time (time $t-1$). This prediction process can be described as follows:

$$P_t = P_{t-1} + \Delta t V_{t-1} \tag{3}$$

$$V_t = \alpha V_{t-1} \tag{4}$$

$$X_t = \begin{bmatrix} 1 & \Delta t \\ 0 & \alpha \end{bmatrix} X_{t-1} \tag{5}$$

where Δt is the time interval, which depends on the data acquisition rate of the Leap Motion, and α is the rate of speed change.

The matrix F_t is used to represent the prediction matrix, so (5) can be represented as follows:

$$X_t = F_t X_{t-1} \tag{6}$$

and through the basic operation of covariance, Y_t can be expressed as the following equation:

$$Y_t = F_t Y_{t-1} F_t^T. \tag{7}$$

3) Refining the Estimate With Measurements: From the

measured sensor data, the current state of the system can be guessed roughly. However, due to uncertainty, some states may be closer to the real state than the measurements acquired from the Leap Motion directly. In this work, covariance R_t is used to express the uncertainty (such as the sensor noise), and the mean value of the distribution is defined as Z_t.

Now, there are two Gaussian distributions, one near the predicted value and the other near the measured value. Therefore, two Gaussian distributions are supposed to be multiplied to calculate the optimal solution between the predicted value and the measured value of the Leap Motion, as shown in the following equations:

$$N(x,\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{8}$$

$$N(x,\mu_0,\sigma_0) \times N(x,\mu_1,\sigma_1) = N(x,\mu',\sigma') \tag{9}$$

where μ_0, σ_0 represent the mean and variance of the predicted values, respectively. μ_1, σ_1 represent the mean and variance of the measured values, respectively. μ', σ' represent the mean and variance of the calculated values, respectively.

By substituting (8) into (9), we can get the following equations:

$$\mu' = \mu_0 + \frac{\sigma_0^2(\mu_1 - \mu_0)}{\sigma_0^2 + \sigma_1^2} \tag{10}$$

$$\sigma'^2 = \sigma_0^2 - \frac{\sigma_0^4}{\sigma_0^2 + \sigma_1^2} \tag{11}$$

the same parts of (10) and (11) are represented by k

$$k = \frac{\sigma_0^2}{\sigma_0^2 + \sigma_1^2}. \tag{12}$$

Therefore, (10) and (11) can be converted as follows:

$$\mu' = \mu_0 + k(\mu_1 - \mu_0) \tag{13}$$

$$\sigma'^2 = \sigma_0^2 - k\sigma_0^2. \tag{14}$$

4) Integrate All the Equations: In this section, the equations of the Kalman filter used in this paper are integrated.

There are two Gaussian distributions, the prediction part

$$(\mu_0, \sigma_0) = (F_t X_t, F_t Y_t F_t^T) \tag{15}$$

and the measurement part

$$(\mu_1, \sigma_1) = (Z_t, R_t). \tag{16}$$

We then put them into (13) and (14) to get the following equation:

$$F_t X_t' = F_t X_t + K(Z_t - F_t X_t) \tag{17}$$

$$F_t Y_t' F_t^T = F_t Y_t F_t^T - K F_t Y_t F_t^T. \tag{18}$$

And according to (12), the Kalman gain is as follows:

$$K = F_t Y_t F_t^T (F_t Y_t F_t^T + R_t)^{-1}. \tag{19}$$

Next, the above three formulas are simplified. On both sides of (17) and (18), we left multiply the inverse matrix of F_t. On both sides of (18), we right multiply the inverse matrix of F_t^T. We then can get the following simplified equation:

$$X_t' = X_t + K'(Z_t - F_t X_t) \tag{20}$$

$$Y_t' = Y_t - K' F_t Y_t \tag{21}$$

$$K' = Y_t F_t^T (F_t Y_t F_t^T + R_t)^{-1}. \tag{22}$$

X_t' is the new optimal estimation of the data collected by the Leap Motion, which we put along with Y_t' into the next prediction and update the equation, and iterate continuously. Through the above steps, the data collected by the Leap Motion could be more accurate.

D. Feature Extraction

Now, after filtering the original data, we analyze four features acquired from the filtered data. These features are introduced in the rest of this section.

- **Coordinate**: x, y, and z coordinates of the hand joints obtained by the Leap Motion.
- **Length**: The distance from the fingertips to center of the hand.
- **Angle**: The angle between the Proximal phalanx and Intermediate phalanx of each finger (except Thumb).
- **Angular velocity**: The rate of the joints angle change.

1) Coordinate Feature: As shown in Fig. 4(a), this feature set represents the position of the finger joints in three-dimensional space. The original data take the Leap Motion as the coordinate origin as shown in Fig. 5. With the movement of the hand, the obtained data could change a lot, which has a certain impact on the experimental results. For the purpose of eliminating the influence of different coordinate systems, the coordinate origin is changed to the palm center, as shown in Fig. 4(a). Taking the palm of the hand as the plane, the direction from palm center to the root of the middle finger is the positive direction of the y-axis. The positive direction of the x-axis is the direction perpendicular to the y-axis and to the right. Through the coordinate origin, perpendicular to this plane is the z-axis.

The positive direction of the y-axis in the new coordinate system can be represented by the following vector:

$$\boldsymbol{A}_y = (x_9 - x_0, y_9 - y_0, z_9 - z_0). \tag{23}$$

Similarly, the positive direction of the x-axis in the new coordinate system can be expressed by the following vector:

$$\boldsymbol{A}_x = (1, \frac{x_0 - x_9}{y_9 - y_0}, 0). \tag{24}$$

And the positive direction of the z-axis in the new coordinate system can be expressed by the following vector:

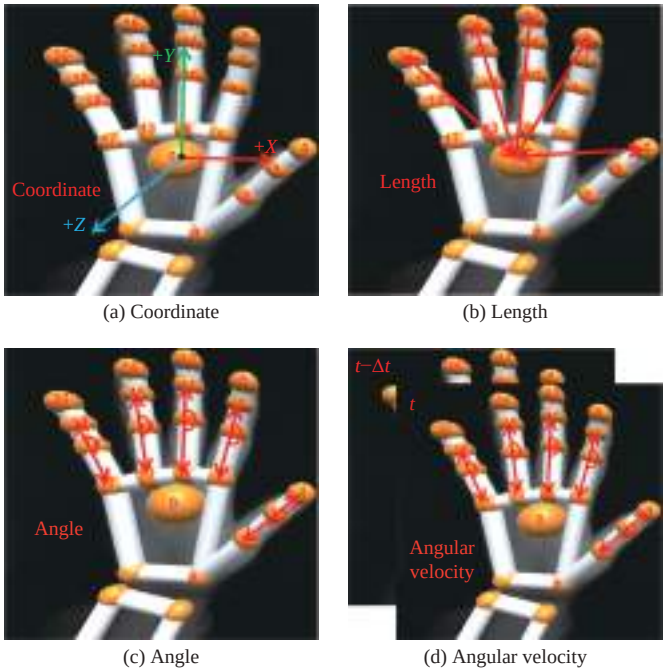
$$\boldsymbol{A}_z = (\frac{(x_9 - x_0)(z_0 - z_9)}{(x_9 - x_0)^2 + (y_9 - y_0)^2}, \frac{(y_9 - y_0)(z_0 - z_9)}{(x_9 - x_0)^2 + (y_9 - y_0)^2}, 1). \tag{25}$$

The coordinate representation in the new coordinate system is

$$d = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2} \tag{26}$$

$$\boldsymbol{A}_i = (x_i - x_0, y_i - y_0, z_i - z_0) \tag{27}$$

$$x_i' = d * \frac{\boldsymbol{A}_i \cdot \boldsymbol{A}_x}{\|\boldsymbol{A}_i\| \times \|\boldsymbol{A}_x\|} \tag{28}$$



(a) Coordinate (b) Length

(c) Angle (d) Angular velocity

Fig. 4. Four different types of features extracted from the Leap Motion.

Fig. 5. The coordinate system of the Leap Motion.

$$y_i' = d * \frac{\boldsymbol{A}_i \cdot \boldsymbol{A}_y}{\|\boldsymbol{A}_i\| \times \|\boldsymbol{A}_y\|} \tag{29}$$

$$z_i' = d * \frac{\boldsymbol{A}_i \cdot \boldsymbol{A}_z}{\|\boldsymbol{A}_i\| \times \|\boldsymbol{A}_z\|} \tag{30}$$

where (x_i', y_i', z_i') represents the new coordinate after coordinate conversion, $i = 1, 2, \ldots, 20$ represent the points corresponding to the finger joints. Through the above equations, we can get new coordinates with the palm center as the origin of the coordinate system. Because each three-dimensional coordinate is a array of length 3, the actual dimension of the coordinate feature is $3 \times 20 = 60$.

2) Length Feature: As shown in Fig. 4(b), this feature refers to the length from each fingertip to the center of the palm. The coordinates of the joints collected from the Leap Motion are used to calculate length information. It can be found that the fingertips are the most variable joints, so (31) is used to calculate the distance between the palm center and the fingertips.

$$L = \sqrt{(x_i' - x_0')^2 + (y_i' - y_0')^2 + (z_i' - z_0')^2} \tag{31}$$

where $i = 4, 8, 12, 16, 20$ represent the points corresponding to the fingertips in Fig. 4(b), and the dimension of length feature is 5.

3) Angle Feature: As shown in Fig. 4(c), this feature represents the angle between Proximal phalanx and Intermediate phalanx of each finger (except Thumb), and the angle extracted from the thumb is between the Intermediate phalanx and Distal phalanx. The calculation process is as follows:

$$\boldsymbol{A}_{i1}' = \boldsymbol{P}_i' - \boldsymbol{P}_s' \tag{32}$$

$$\boldsymbol{A}_{i2}' = \boldsymbol{P}_j' - \boldsymbol{P}_s' \tag{33}$$

$$\alpha = \arccos\left(\frac{\boldsymbol{A}_{i1}' \cdot \boldsymbol{A}_{i2}'}{\|\boldsymbol{A}_{i1}'\| \times \|\boldsymbol{A}_{i2}'\|}\right) \tag{34}$$

where $\boldsymbol{P}_i', \boldsymbol{P}_j', \boldsymbol{P}_s'$ represent the three-dimensional position of finger joint in the new coordinate system, $i = 4, 7, 11, 15, 19$, $j = 2, 5, 9, 13, 17$, $s = 3, 6, 10, 14, 18$. The dimension of angle feature is 5.

4) Angular Velocity Feature: As shown in Fig. 4(d), this feature represents the rate of the joint angle change. As shown in the following equation:

$$\omega = \frac{\alpha_t - \alpha_{t-\Delta t}}{\Delta t} \tag{35}$$

where t is the current time, Δt is the time interval, which depends on the Leap Motion's sampling time. The dimension of the angular velocity feature is 5.

E. Gesture Prediction

The method in the previous section produces four different features, and each feature represents some information related to the performed gesture. In this section, the LSTM network [24] used for gesture prediction is described in detail. The internal structure of the LSTM is shown in Fig. 6, where x_t denotes the input of the LSTM network and h_t denotes the output of the LSTM network. f_t is the forget gate variables, i_t is the input gate variables, and o_t is the output gate variables. The subscripts t and $t-1$ represent the current time and previous time. c_t and $\widetilde{c}_t$ are the memory cell state and the memory gate, respectively. The notation of σ_{lstm} and *tanh* denote the sigmoid and hyperbolic activation functions as shown in (36) and (37).

$$\sigma_{\text{lstm}}(x) = \frac{1}{1 + e^{-x}} \tag{36}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{37}$$

The relevant parameters of the LSTM can be calculated by the following equations:

$$f_t = \sigma_{\text{lstm}}(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{38}$$

$$i_t = \sigma_{\text{lstm}}(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{39}$$

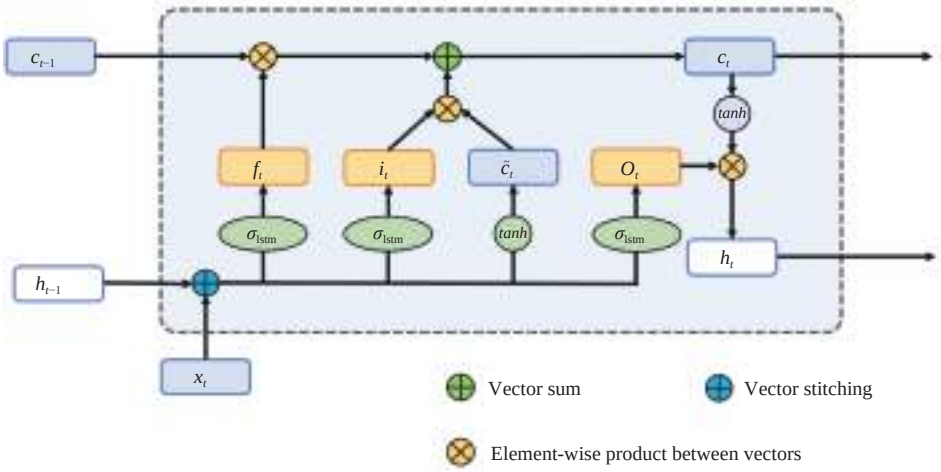


Fig. 6. The internal structure of the LSTM.

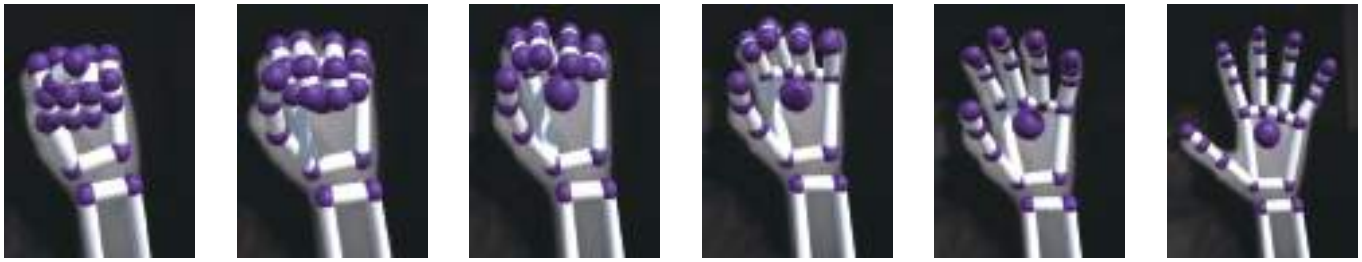


Fig. 7. Collect gesture paper data by using the Leap Motion.

$$\widetilde{c}_t = tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{40}$$

$$c_t = f_t * c_{t-1} + i_t * \widetilde{c}_t \tag{41}$$

$$o_t = \sigma_{\text{lstm}}(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{42}$$

$$h_t = o_t * tanh(c_t) \tag{43}$$

where subscripts of f, i, o, and c are related to the parameters of the forget gate, input gate, output gate and memory cell. The parameters W_f, W_i, W_c, and W_o denote the weight matrices of the corresponding subscripts. Similarly, b_f, b_i, b_c, and b_o present the biases corresponding to subscripts of the LSTM network. The notation of $*$ denotes the element-wise product between vectors.

In the process of data collection, the change of gestures can be divided into three stages, as shown in Fig. 7. For a clearer description, we take the process of turning rock into paper as an example to explain these three stages:

1) The Original Stage: As shown in Figs. 7(a) and 7(b), the gestures at this stage are close to the original state, that is, the gesture is similar to a rock.

2) The Intermediate Stage: As shown in Figs. 7(c) and 7(d), the gestures at this stage change significantly compared to the original stage, that is, the five fingers clearly show different degrees of openness.

3) The Completion Stage: As shown in Figs. 7(e) and 7(f), the gestures at this stage are close to the completion, that is, the gestures tend to being paper.

Since different players perform actions at different speeds, each action contains 2–6 frames. For the purpose of uniformity, the T of LSTM is set to 4, that is, 4 frames of data are input into the LSTM network for prediction. This process is shown in Fig. 8. The input layer of the LSTM network is features obtained by the Leap Motion. These features are the coordinate, length, angle and angular velocity calculated from Section II-D, and their dimensions are 60, 5, 5, and 5, respectively. In addition, the hidden layer of the LSTM network contains 100 nodes. The output of the LSTM network is the result of gesture prediction with the dimension of 3, that is, rock, scissors and paper. With the LSTM network, we can predict the gestures accurately, and the classification results will be sent to a social robot for interaction and reaction.

III. Experiment

A. Experimental Setup

In this section, the performance and efficiency of the proposed framework are tested. The experiments were carried out on a laptop with an Intel Core i5-6200U CPU. The dynamic gestures of rock, paper and scissors are collected from five different players, and each player repeats each gesture 300 times at fast, medium, and slow speeds, for a total of 4500 different data samples. The experimental results of the network trained by the four features and their combination are compared.

B. Kalman Filter

In Section II-C, the Kalman filter is introduced in detail. In this section, it is verified by an experiment, and the measured

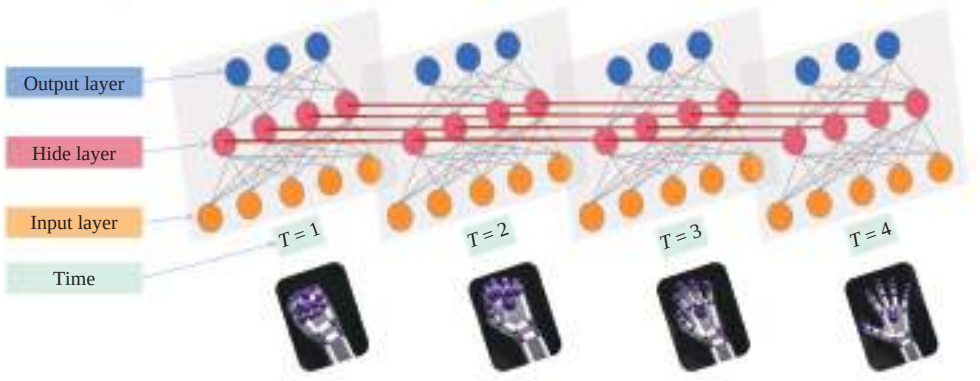


Fig. 8. The process of the LSTM for predicting gestures.

position is directly obtained by the Leap Motion. The Kalman filter is used to process the original coordinate data to make the processed data closer to the real value. As can be seen from Fig. 9, the processed data is much smoother.

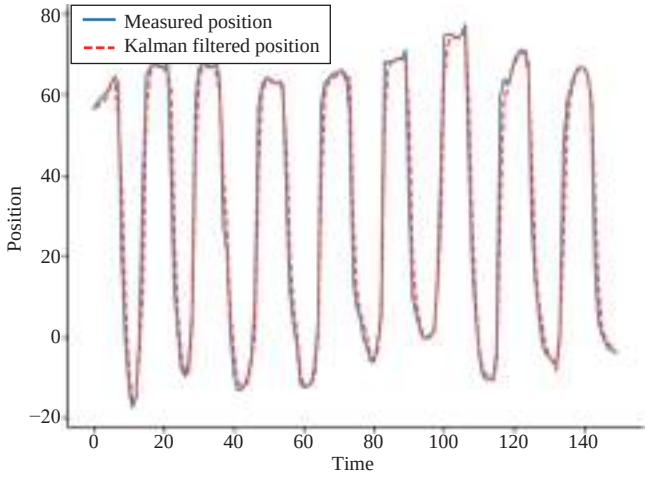


Fig. 9. Data processed by Kalman filter.

C. Experimental Result

According to the description in Section II-D, we extract the three-dimensional coordinates feature, length feature, angle feature, and angular velocity feature from the filtered data, and train them. Figs. 10 and 11 show the accuracy of features using the classification algorithm of Section II-E.

The three-dimensional positions of the finger joints show that the accuracy of gesture prediction is 97.93%. The length feature and the angle feature have an accuracy of 95.17% and 93.79%, respectively. The angular velocity feature has lower performance, it has an accuracy of 79.31%. It is affected by the speed of the player's movement, so it is not fast enough to make an accurate prediction.

The combination of multiple features could enrich the input of the neural network. In some cases, it maybe improve the performance of the prediction. As can be seen from Fig. 11, the combination of coordinate features, length features and angle features achieve the highest accuracy of 99.31%, better than any of the three features alone. These results suggest that different features can represent different attributes of the hand

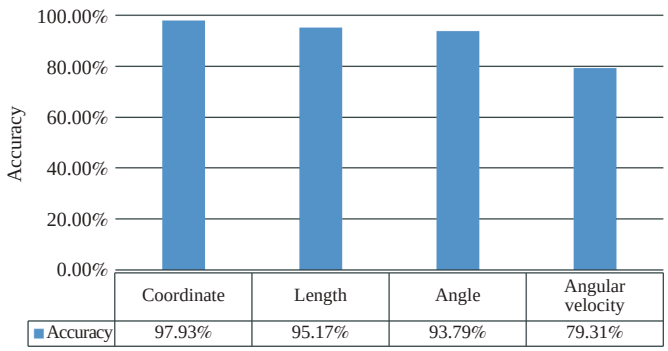


Fig. 10. The experimental results of four features.

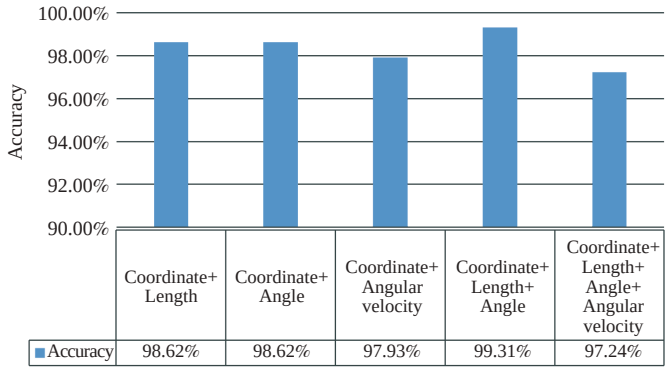


Fig. 11. The experimental results of the combination of four features.

and include complementary information.

We examine whether the proposed method is able to achieve real-time gesture recognition and prediction. As shown in Figs. 12 and 13, it is obvious that the method proposed in this work can predict the gesture of the finger-guessing game very well. For example, when the player's gesture changes from rock to paper, the proposed method can predict that the player's gesture is paper before all fingers are fully open. In addition, we also verify the prediction results of the proposed method from different angles of the Leap Motion to the hand, as shown in Fig. 14.

D. Application

In order to further prove the effectiveness of the proposed

Fig. 12. The prediction process of turning rock into paper.

Fig. 13. The prediction process of turning rock into scissors.

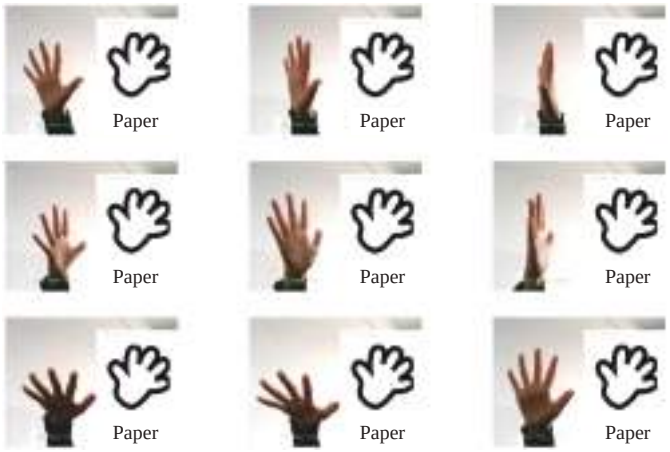


Fig. 14. Predicted results from different angles of the Leap Motion to hand.

Fig. 15. Experimental equipment and platform.

method, the trained network is applied to the humanoid robot NAO, as shown in Fig. 15. The NAO is an autonomous, programmable humanoid robot which is designed by Aldebaran Robotics [32]. The height of the NAO is 573.2 mm and the weight of it is 4.5 kg. It has two cameras, voice recognition, voice synthesis and powered by LiPo Battery. What's more, it consists of four microphones, two sonar emitters and receivers, two IR emitters and receivers, and three tactile sensors on the top of head.

In this work, we mainly use the NAO robot's left hand to play the finger-guessing game with the player. As shown in Fig. 16, the NAO robot has only three fingers, and they are linked. Therefore, we first define that the full opening of the robot fingers is paper, the half opening of the robot fingers is scissors, and the clenched fingers are rock.

Then, the trained model is applied to the NAO robot, and the experimental results are shown in Fig. 17. The Leap Motion is used to predict gestures, and then the computer sends the results to the NAO robot, so that the NAO robot can win or lose the game through some simple judgments.

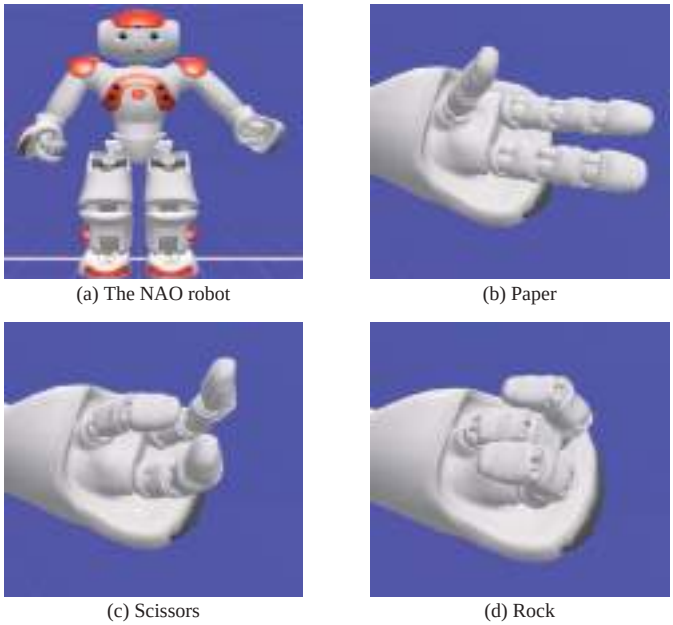
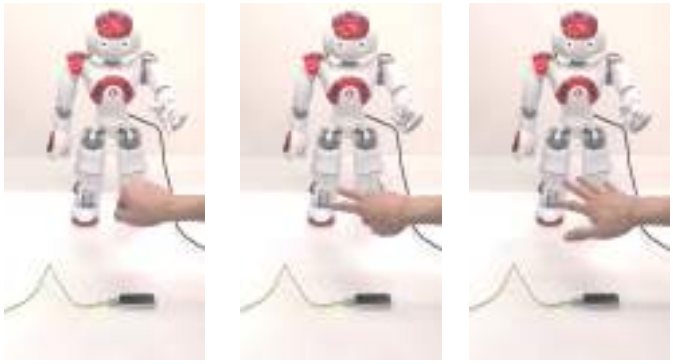


Fig. 16. The NAO robot and rock-paper-scissors gesture.

IV. Conclusion

In this paper, a gesture prediction framework based on the Leap Motion is proposed. In the process of data acquisition by the Leap Motion, some jumps or jitters maybe occur.



(a) (b) (c)

Fig. 17. The experimental results with the NAO robot.

Therefore, the Kalman filter is used to solve these problems. Then, based on the original coordinate features collected by the Leap Motion, we extract three new features, namely, the length feature, angle feature and angular velocity feature. The LSTM network is used to train the model for gesture prediction. In addition, the trained model is applied to the NAO robot to verify the real-time and effectiveness of the proposed method.

REFERENCES

[1] C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C.-Y. Su, "Mind control of a robotic arm with visual fusion technology," *IEEE Trans. Industrial Informatics*, vol. 14, no. 9, pp. 3822–3830, 2017.

[2] J. Zhang, M. Li, Y. Feng, and C. Yang, "Robotic grasp detection based on image processing and random forest," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 2427–2446, 2020.

[3] J. Li, J. Zhong, J. Yang, and C. Yang, "An incremental learning framework to enhance teaching by demonstration based on multimodal sensor fusion," *Frontiers in Neurorobotics*, vol. 14, p. 5, 2020.

[4] P. Premaratne, "Historical development of hand gesture recognition," in *Human Computer Interaction Using Hand Gestures*. Singapore: Springer, 2014, pp. 5–29.

[5] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *Journal of Imaging*, vol. 6, no. 8, p. 73, 2020.

[6] A. S. Al-Shamayleh, R. Ahmad, M. A. Abushariah, K. A. Alam, and N. Jomhari, "A systematic literature review on vision based gesture recognition techniques," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28121–28184, 2018.

[7] L. Dipietro, A. M. Sabatini, and P. Dario, "A survey of glove-based systems and their applications," *IEEE Trans. Systems*, *Man*, *and Cybernetics*, *Part C* (*Applications and Reviews*), vol. 38, no. 4, pp. 461–482, 2008.

[8] A. Jaramillo-Yánez, M. E. Benalcázar, and E. Mena-Maldonado, "Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review," *Sensors*, vol. 20, no. 9, p. 2467, 2020.

[9] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao, "Surface EMG hand gesture recognition system based on PCA and GRNN," *Neural Computing and Applications*, vol. 32, no. 10, pp. 6343–6351, 2020.

[10] S. Jiang, Q. Gao, H. Liu, and P. B. Shull, "A novel, co-located EMG-FMG-sensing wearable armband for hand gesture recognition," *Sensors and Actuators A: Physical*, vol. 301, p. 111738, 2020.

[11] W.-T. Shi, Z.-J. Lyu, S.-T. Tang, T.-L. Chia, and C.-Y. Yang, "A bionic hand controlled by hand gesture recognition based on surface EMG signals: A preliminary study," *Biocybernetics and Biomedical Engineering*, vol. 38, no. 1, pp. 126–135, 2018.

[12] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2015.

[13] A. Kurakin, Z. Zhang, and Z. Liu, "A real time system for dynamic hand gesture recognition with a depth sensor," in *Proc. IEEE 20th European Signal Processing Conf.*, 2012, pp. 1975–1979.

[14] H. Cheng, L. Yang, and Z. Liu, "Survey on 3D hand gesture recognition," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 26, no. 9, pp. 1659–1673, 2015.

[15] P. Barros, N. T. Maciel-Junior, B. J. Fernandes, B. L. Bezerra, and S. M. Fernandes, "A dynamic gesture recognition and prediction system using the convexity approach," *Computer Vision and Image Understanding*, vol. 155, pp. 139–149, 2017.

[16] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Computer Vision and Image Understanding*, vol. 141, pp. 152–165, 2015.

[17] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[18] Y. Li, "Hand gesture recognition using kinect," in *Proc. IEEE Int. Conf. Computer Science and Automation Engineering*, 2012, pp. 196–199.

[19] D. Bachmann, F. Weichert, and G. Rinkenauer, "Review of three-dimensional human-computer interaction with focus on the leap motion controller," *Sensor*s, vol. 18, no. 7, p. 2194, 2018.

[20] D. Avola, M. Bernardi, L. Cinque, G. L. Foresti, and C. Massaroni, "Exploiting recurrent neural networks and leap motion controller for the recognition of sign language and semaphoric hand gestures," *IEEE Trans. Multimedia*, vol. 21, no. 1, pp. 234–245, 2018.

[21] Z. Zhang, C. He, and K. Yang, "A novel surface electromyographic signal-based hand gesture prediction using a recurrent neural network," *Sensors*, vol. 20, no. 14, p. 3994, 2020.

[22] C. Wei, W. Zhou, J. Pu, and H. Li, "Deep grammatical multi-classifier for continuous sign language recognition," in *Proc. IEEE 5th Int. Conf. Multimedia Big Data*, 2019, pp. 435–442.

[23] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "A multimodal framework for sensor based sign language recognition," *Neurocomputing*, vol. 259, pp. 21–38, 2017.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, no. 9, pp. 1735–1780, 1997.

[25] X. Yu, W. He, H. Li, and J. Sun, "Adaptive fuzzy full-state and outputfeedback control for uncertain robots with output constraint," *IEEE Trans. Systems*, *Man*, *and Cybernetics*: *Systems*, 2020.

[26] W. He, Z. Li, and C. P. Chen, "A survey of human-centered intelligent robots: Issues and challenges," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 602–609, 2017.

[27] T.-W. Chong and B.-G. Lee, "American sign language recognition using leap motion controller with machine learning approach," *Sensors*, vol. 18, no. 10, p. 3554, 2018.

[28] A. Vaitkevičius, M. Taroza, T. Blažauskas, R. Damaševičius, R. Maskeliūnas, and M. Woźniak, "Recognition of american sign language gestures in a virtual reality using leap motion," *Applied Sciences*, vol. 9, no. 3, p. 445, 2019.

[29] W. Zeng, C. Wang, and Q. Wang, "Hand gesture recognition using leap motion via deterministic learning," *Multimedia Tools and Applications*,

vol. 77, no. 21, pp. 28185–28206, 2018.

[30] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with jointly calibrated leap motion and depth sensor," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 1–25, 2016.

[31] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.

[32] D. Gouaillier, V. Hugel, P. Blazevic, C. Kilner, J. Monceaux, P. Lafourcade, B. Marnier, J. Serre, and B. Maisonnier, "Mechatronic design of NAO humanoid," in *Proc. IEEE Int. Conf. Robotics and Automation*, 2009, pp. 769–774.

Bixiao Wu received the B.Eng. degree in electrical engineering from the Soochow University in 2019. She is currently pursuing the M.Sc. degree in the South China University of Technology. Her research interests include human robot interaction, gesture recognition and transfer learning.

Junpei Zhong (Senior Member, IEEE) received the B.Eng degree in control science and computer science from South China University of Technology in 2006, the M.Phil degree in electrical engineering from Hong Kong Polytechnic University, China, in 2010, and the Ph.D. degree in computer science from University of Hamburg, Germany, in 2015. He has been awarded the Marie-Curie fellowship for his doctoral study from 2010 to 2013. From 2014 to 2016, he has participated in different European Union and Japanese funded projects at University of Hertfordshire, UK, Plymouth University, UK, and Waseda University, Japan. His research interests include machine learning, computational intelligence and cognitive robotics.

Chenguang Yang (Senior Member, IEEE) received the B.Eng. degree in measurement and control from Northwestern Polytechnical University in 2005, the Ph.D. degree in control engineering from the National University of Singapore, Singapore, in 2010, and postdoctoral training in human robotics from the Imperial College London, UK. His research interests include robotics and automation. Dr Yang was a Recipient of the IEEE Transactions on Robotics Best Paper Award (2012) and IEEE Transactions on Neural Networks and Learning Systems Outstanding Paper Award (2022) as leading author.