

Surface Electromyography (sEMG)-based Intention Recognition and Control Design for Human–Robot Interaction in Uncertain Environment

Junbao Gan,¹ Ning Wang,^{2*} and Lei Zuo³

¹Key Laboratory of Autonomous Systems and Networked Control, College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China

²Bristol Robotics Laboratory, University of the West of England, Frenchay, Coldharbour Ln, Bristol BS34 8QZ, UK

³School of Electronic and Control Engineering, Chang'an University, Xi'an 710064, China

(Received December 20, 2020; accepted April 16, 2021)

Keywords: human–robot interaction, surface electromyography, barrier Lyapunov function, radial basis function neural network

An important direction of human–robot interaction (HRI) is making robots respond to complex and dexterous tasks intelligently. To achieve this, biological signals based on surface electromyography (sEMG) have widely been used to identify human intentions rapidly and effectively. We propose an algorithm that can recognize human intentions conveyed by different hand gestures through analyzing sEMG data. This will facilitate the selection of the most appropriate interaction mode and level during HRI for the robot. We also propose an admittance control framework combining a tan-type barrier Lyapunov function (BLF) and a radial basis function neural network (RBFNN) to ensure the interaction and tracking performance and to guarantee the stability of the system in uncertain environments. Experiments performed on a Baxter robot verify the effectiveness of the proposed framework.

1. Introduction

Nowadays, robots are widely used in industrial automation, rehabilitation therapy, surgery, and daily life,^(1–5) and inevitably interact with the external environment in actual application. It is essential to ensure the intelligent response of robots to complete more complex, dexterous, and multifunctional tasks in an unstructured environment. Therefore, human–robot interaction (HRI) has become a research hotspot.

Compliant control and tracking control are used for intelligent HRI. A lot of meaningful work on compliant control has been carried out, and two mainstream methods are hybrid position/force control and impedance control.^(6–9) Without considering the dynamic coupling between environment and robot, the accuracy of hybrid position/force control cannot be guaranteed. In contrast, impedance control aims to adjust the mechanical impedance to the target impedance, which has better robustness and safety. Impedance control is mainly divided

*Corresponding author: e-mail: Katie.Wang@brl.ac.uk
<https://doi.org/10.18494/SAM.2021.3230>

into impedance control and admittance control according to the causality of the controller. Owing to the inertia and friction of robots, impedance control is usually suitable for dynamic interaction in a rigid environment. Admittance control can obtain the reference trajectory from the required admittance model and measured external force, and is mostly used in flexible environments.^(10,11) Therefore, we apply admittance control to deal with the interaction with a flexible environment. Accurate kinematic information of a robot is usually provided by the manufacturer, but the dynamics of a robot is uncertain in actual situations, which will have a negative effect on the performance of tracking control. However, the available input and output data could be used to approximate unknown robot dynamics. Thus, a controller based on a radial basis function neural network (RBFNN) is widely used.^(12–14) Moreover, the position and the speed may change markedly upon interaction owing to low control accuracy, so the transient performance should be considered for tracking control. Many studies utilized a barrier Lyapunov function (BLF) to change the transient performance of the control system.^(15–18) Li *et al.* developed a robust adaptive fault-tolerant control method based on the BLF to ensure transient performance and robustness against actuator failure.⁽¹⁵⁾ Considering the internal force between dual arms, Yang *et al.* used the BLF to specify the tracking performance in transient and steady states to achieve effective cooperation between dual arms.⁽¹⁶⁾ Sun *et al.* combined the time-varying BLF technology and a high-gain feedback method to limit the output tracking error in a predefined arbitrary area.⁽¹⁷⁾

One of the most critical problems in HRI is making robots understand the movement intention of humans. Common intention recognition techniques include the use of vision, natural language processing (NLP), and biological signals.^(19–24) The visual method can easily produce incorrect judgments when the view is blocked by an object, whereas for the NLP method, industrial applications are limited by the need for voice acquisition and text input. However, biological-based intention recognition methods usually capture the operator's movement intention rapidly and effectively. A surface electromyography (sEMG) signal is one of the most widely used biological signals for interaction. Kong *et al.* proposed a shared control strategy that used sEMG signals to control the direction and speed of mobile robots.⁽²⁵⁾ To obtain adaptive impedance control effectively, Zeng *et al.* used sEMG to extract the characteristics of human arm stiffness and then mapped the estimated stiffness to an impedance controller.⁽²⁶⁾ Yang *et al.* proposed a variable gain control mechanism to make a remote operating system naturally interact with the external environment by utilizing a task learning framework and the recorded sEMG signal.⁽²⁷⁾ In all these studies, sEMG signals were used to achieve more intelligent HRI. However, the force applied to the same object may be different in different situations, which may cause the falling of objects in HRI.

In this paper, we propose an algorithm to make the robot automatically select the interaction mode and level during HRI by recognizing human intentions through their hand gestures. A BLF-based RBFNN control framework is applied to a constraint manipulator, which interacts with an unknown environment. Compared with existing research, the main contributions of this paper are as follows.

- (1) We propose an algorithm for human intention recognition that can automatically select the interaction mode and level according to the classification results of hand gestures.

(2) We present a control framework combining a tan-type BLF and an RBFNN that can limit the tracking error within a certain range and achieve good transient performance during HRI.

The rest of this paper is organized as follows. In Sect. 2, related preliminary research is described. In Sect. 3, the method of human intention recognition and the design of the controller with its proof of stability are introduced. In Sect. 4, the effectiveness of our proposed method is verified through experiments conducted on a Baxter robot. Section 5 shows conclusions.

2. Preliminaries

2.1 Robot dynamic description

Murray *et al.*⁽²⁸⁾ have derived the formulas for robot dynamics in detail, and the following formulas and properties are based on their work. The dynamics of a robot in the joint space is

$$\mathbf{M}_q(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}_q(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}_q(\mathbf{q}) = \mathbf{u} + \mathbf{f}, \quad (1)$$

where \mathbf{q} , $\dot{\mathbf{q}}$, and $\ddot{\mathbf{q}}$ are the coordinate, speed, and acceleration in the joint space, respectively. $\mathbf{M}_q(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the inertia matrix of the manipulator, $\mathbf{C}_q(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ represents the centrifugal and Coriolis force vector, $\mathbf{G}_q(\mathbf{q}) \in \mathbb{R}^{n \times 1}$ represents the gravity torques, $\mathbf{u} \in \mathbb{R}^n$ is the vector of the control input, and $\mathbf{f} \in \mathbb{R}^n$ is the vector of the measured interaction force, which is the force exerted by the human on the robotic arm.

Since the manipulator is controlled in Cartesian space, the above formula is transformed from the joint space to Cartesian space as

$$\mathbf{M}_x(\mathbf{q})\ddot{\mathbf{x}} + \mathbf{C}_x(\mathbf{q}, \dot{\mathbf{x}})\dot{\mathbf{x}} + \mathbf{G}_x(\mathbf{q}) = \boldsymbol{\tau}_u + \boldsymbol{\tau}_f, \quad (2)$$

where \mathbf{x} , $\dot{\mathbf{x}}$, and $\ddot{\mathbf{x}}$ are the coordinate, speed, and acceleration in Cartesian space, respectively. The remaining variables are transformed from the joint space to Cartesian space, and the robot dynamics have two properties as follows.

$$\mathbf{M}_x = \mathbf{J}^{-T} \mathbf{M}_q \mathbf{J}^{-1} \quad (3)$$

$$\mathbf{C}_x = \mathbf{J}^{-T} (\mathbf{C}_q - \mathbf{M}_q \mathbf{J}^{-1} \dot{\mathbf{J}}) \mathbf{J}^{-1} \quad (4)$$

$$\mathbf{G}_x = \mathbf{J}^{-T} \mathbf{G}_q \quad (5)$$

$$\boldsymbol{\tau}_u = \mathbf{J}^{-T} \mathbf{u} \quad (6)$$

$$\boldsymbol{\tau}_f = \mathbf{J}^{-T} \mathbf{f} \quad (7)$$

Property 1: $\mathbf{M}_x(\mathbf{q})$ is a symmetric and positive definite matrix.

Property 2: $2\mathbf{C}_x(\mathbf{q}, \dot{\mathbf{q}}) - \dot{\mathbf{M}}_x(\mathbf{q})$ is a skew-symmetric matrix.

2.2 RBF neural network

An RBFNN is widely used to deal with uncertainties in various systems because of its universal approximation of⁽²⁹⁾

$$\boldsymbol{\varphi}(\mathbf{Z}) = \mathbf{W}^{*T} \mathbf{S}(\mathbf{Z}), \quad (8)$$

where $\boldsymbol{\varphi}(\mathbf{Z}) \in \mathbb{R}^{n \times 1}$, $\mathbf{W}^{*T} \in \mathbb{R}^{n \times k}$ denotes the ideal weight matrix, which cannot be obtained in practice, $\mathbf{Z} \in \mathbb{R}^{m \times 1}$ is the input vector, and $\mathbf{S}(\mathbf{Z}) \in \mathbb{R}^{k \times 1}$ is a vector composed of radial basis functions. In general, we select the Gaussian function as the radial basis function, and the i th $\mathbf{S}(\mathbf{Z})$ is calculated as

$$S_i = \exp\left(-\|\mathbf{Z} - \mathbf{u}_i\|^2 / \Omega_i^2\right), \quad (9)$$

where $\mathbf{u}_i \in \mathbb{R}^{m \times 1}$ is the central vector and Ω_i is the width of the radial basis function. The dynamic parameters of the manipulator are

$$\begin{aligned} \mathbf{M}_x &= \mathbf{W}_M^{*T} \mathbf{S}_M, \\ \mathbf{C}_x &= \mathbf{W}_C^{*T} \mathbf{S}_C, \\ \mathbf{G}_x &= \mathbf{W}_G^{*T} \mathbf{S}_G. \end{aligned} \quad (10)$$

$\hat{\mathbf{M}}_x$, $\hat{\mathbf{C}}_x$, and $\hat{\mathbf{G}}_x$ are the dynamic parameters of the manipulator approximated using the RBFNN in Eq. (8),

$$\begin{aligned} \hat{\mathbf{M}}_x &= \hat{\mathbf{W}}_M^T \mathbf{S}_M + \boldsymbol{\varepsilon}_M, \\ \hat{\mathbf{C}}_x &= \hat{\mathbf{W}}_C^T \mathbf{S}_C + \boldsymbol{\varepsilon}_C, \\ \hat{\mathbf{G}}_x &= \hat{\mathbf{W}}_G^T \mathbf{S}_G + \boldsymbol{\varepsilon}_G, \end{aligned} \quad (11)$$

where $\boldsymbol{\varepsilon}_M$, $\boldsymbol{\varepsilon}_C$, and $\boldsymbol{\varepsilon}_G$ represent the bounded approximation errors of the neural network. We define the deviation signal of the RBFNN is equal to estimated value minus actual value, which would not affect the stability analysis.

$$\begin{aligned} \tilde{\mathbf{M}}_x &= \tilde{\mathbf{W}}_M^T \mathbf{S}_M + \boldsymbol{\varepsilon}_M \\ \tilde{\mathbf{C}}_x &= \tilde{\mathbf{W}}_C^T \mathbf{S}_C + \boldsymbol{\varepsilon}_C \\ \tilde{\mathbf{G}}_x &= \tilde{\mathbf{W}}_G^T \mathbf{S}_G + \boldsymbol{\varepsilon}_G \end{aligned} \quad (12)$$

3. Human Intention Recognition and Controller Design

3.1 Human intention recognition based on sEMG signal

sEMG is an important feature for the non-invasive recognition of muscle activity. Different intensities of an sEMG signal can represent different muscle movements. To select sEMG features initially before preprocessing, we obtain eight channels of hand gestures and use a 320-ms-long sliding window to intercept 64 frames of data for each channel as the feature of one sampling. Human intention recognition is based on the following structure including an offline training process and an online intention estimate, as shown in Fig. 1. The training process is offline, and the features of the raw sEMG signals of different hand gestures are extracted through a low-pass filter and a sliding window. In practice, the feature of the raw sEMG signal is extracted rapidly and then imported as an input to a convolutional neural network (CNN) classifier for hand gesture recognition online.

We set up six hand gestures to represent three interaction modes and three interaction levels as shown in Fig. 2. Among them, gestures 1 to 3 represent the pause, forward, and backward modes, respectively. Gestures 4 to 6 represent levels 1, 2, and 3, respectively. In the pause mode, the robot stops its motion to ensure safety during HRI until the next hand gesture is recognized. In the forward mode, the human pushes the robot to move in the forward direction, while in the backward mode, we add a virtual force to the manipulator to keep the robot following the human

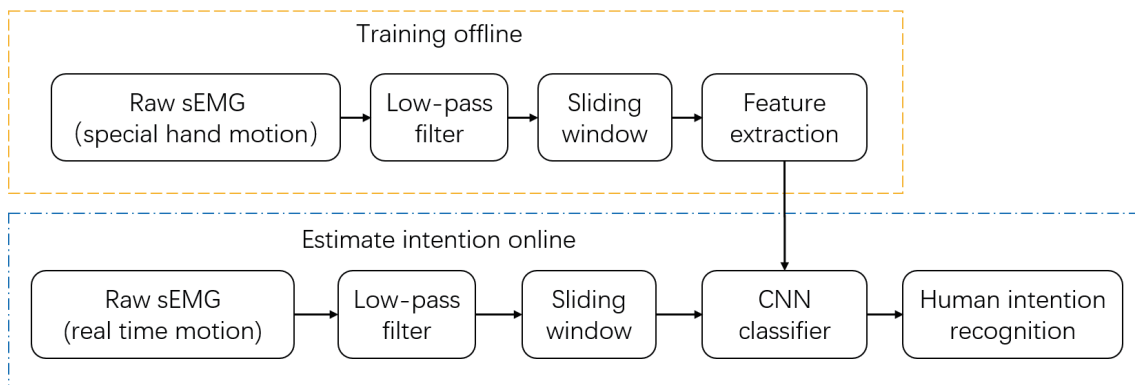


Fig. 1. (Color online) Training offline and estimate intention online of sEMG signal.

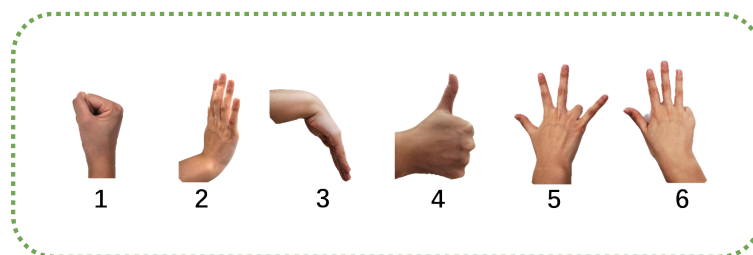


Fig. 2. (Color online) Hand gestures associated with three interaction modes (gestures 1–3) and three interaction levels (gestures 4–6).

in the backward direction. These two modes represent common back-and-forth motion during HRI. However, when the human and robot hold an object collectively in the backward mode, the object is more likely to fall if the robot cannot follow the human motion with the same pace. To solve this problem, we set different interaction levels. A higher interaction level means a larger force will be exerted on the object by the robot. In this way, the interaction levels, i.e., the forces exerted by the robot, are determined according to the manipulation task, e.g., the weight of the ball in this example. Although this way of setting interaction levels applies to both the forward and backward modes, the levels for these two modes are independently determined.

We propose an algorithm that can automatically switch the interaction mode and level by human intention recognition. Algorithm 1 shows the pseudocode for achieving this process. In Algorithm 1, we define two flags: *flag1* represents the interaction mode and *flag2* represents the interaction level. We set three levels for *flag2*, so that it can cope with three objects with different weights. These levels, which represent the interaction force, can easily be changed according to the needs in other HRI applications. The threshold setting is related to the additional sEMG signal activation value when we change the hand gesture, and it is set to 30% of the maximum sEMG activation value of the minimum-intensity exercise. As also shown in Algorithm 1, when an sEMG signal reaches the set threshold, it is fed to a CNN classifier to produce the classification result. When the first gesture is identified successfully, the robot will move accordingly, e.g., when gesture 1 is identified, the robot will stop and enter the pause mode to ensure safety. Otherwise, if the classification result is either gesture 2 or 3, *flag1* will be changed to choose either the forward or backward mode. After setting *flag1*, we need to define *flag2*. Now we use gestures 4, 5, and 6 to define *flag2*. When the second hand gesture is identified, we choose the interaction level accordingly by changing *flag2*. The robot will then switch to the appropriate interaction mode and level to facilitate the physical interaction with the human partner.

Algorithm 1

Framework of automatically switching interaction mode and level.

Input: Different sEMG signals

Output: the interaction mode *flag1* and level *flag2* which control robot motion

```

1: flag1 = 0, flag2 = 0
2: while not receive end signal
3:   accept and update sEMG signal
4:   if  $\sqrt{\sum sEMG^2} \geq \text{Threshold}$ 
5:     input to CNN and obtain sEMG signal classification result  $\kappa$ 
6:     if  $\kappa \in \{\text{gesture 1}\}$ 
7:       mode = pause mode
8:       continue
9:     else if  $\kappa \in \{\text{gesture 2, gesture 3}\}$ 
10:      flag1 = corresponding mode of  $\kappa$ ,  $\text{mode} \in \{\text{forward mode, backward mode}\}$ 
11:      continue
12:     else if flag1 ≠ 0 and  $\kappa \in \{\text{gesture 4, gesture 5, gesture 6}\}$ 
13:      flag2 = the corresponding level of  $\kappa$ ,  $\text{level} \in \{\text{level 1, level 2, level 3}\}$ 
14:      send flag1 and flag2 to robot
15:     else
16:       goto step 3

```

3.2 Design of RBFNN controller based on BIF

We design the RBFNN controller based on the BLF and introduce the design process and stability analysis in detail in this section. Figure 3 shows the general framework of our proposed controller. The Cartesian tracking error is defined as

$$\begin{cases} z_1 = \mathbf{x} - \mathbf{x}_d, \\ z_2 = \dot{\mathbf{x}} - \boldsymbol{\alpha}, \end{cases} \quad (13)$$

where $\mathbf{z}_1 = [z_{11} \ \dots \ z_{1n}]^T$, $\mathbf{x}_d = [x_{d1} \ \dots \ x_{dn}]^T$ is the desired position in Cartesian space, and $\boldsymbol{\alpha}$ is the virtual controller, which can be designed as

$$\boldsymbol{\alpha} = \dot{\mathbf{x}}_d - \mathbf{A}, \quad (14)$$

where $\mathbf{A} = \begin{bmatrix} k_{11}z_{11} \cos\left(\frac{\pi z_{11}^2}{2a_1^2}\right) \\ \vdots \\ k_{1i}z_{1i} \cos\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) \end{bmatrix}$, $k_{1i} > 0, i = 1, \dots, n$.

Let $\mathbf{x}_1 = \mathbf{x}$ and $\mathbf{x}_2 = \dot{\mathbf{x}}$, then Eq. (2) can be transformed into

$$\begin{cases} \dot{\mathbf{x}}_1 = \mathbf{x}_2, \\ \dot{\mathbf{x}}_2 = \mathbf{M}_x^{-1}(\boldsymbol{\tau}_u + \boldsymbol{\tau}_f - \mathbf{C}_x \dot{\mathbf{x}} - \mathbf{G}_x). \end{cases} \quad (15)$$

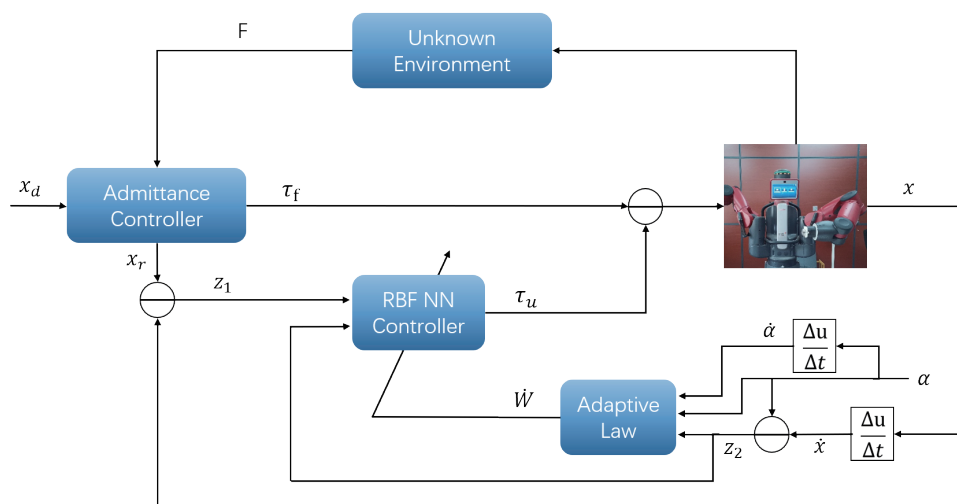


Fig. 3. (Color online) Structure of RBFNN controller.

The robot interacts with the unknown environment through admittance control, and the adaptive control torque τ_u is

$$\begin{aligned} \tau_u = & -\mathbf{k}_p \mathbf{z}_2 - (\mathbf{z}_2^T)^+ \sum_1^n \frac{k_{1i} a_i^2}{\pi} \tan\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) - \sum_1^n z_{1i} \sec\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) \\ & - \mathbf{k}_r \text{sgn}(z_2) - \tau_f + \hat{\mathbf{M}}_x \dot{\boldsymbol{\alpha}} + \hat{\mathbf{C}}_x \boldsymbol{\alpha} + \hat{\mathbf{G}}_x, \end{aligned} \quad (16)$$

where $\mathbf{k}_p > 0$, $(\mathbf{z}_2^T)^+$ is the pseudoinverse of error \mathbf{z}_2 , $\text{sgn}(\ast)$ is the sign function, and $\mathbf{k}_r > 0$ with $\mathbf{k}_r > \|\boldsymbol{\varepsilon}_M \dot{\boldsymbol{\alpha}} + \boldsymbol{\varepsilon}_C \boldsymbol{\alpha} + \boldsymbol{\varepsilon}_G\|$.

3.3 Stability analysis

In this part, we utilize the Lyapunov method to strictly evaluate the stability of the control system and derive the stable adaptive update law of the RBFNN in detail. Owing to the low control accuracy, the position and speed may change markedly during an interaction, resulting in a strong collision and unpredictable results. Therefore, we introduce a BLF⁽³⁰⁾ to solve this problem:

$$V(t) = \frac{a^2}{\pi} \tan\left(\frac{\pi z^2}{2a^2}\right), \quad (17)$$

where z is the system state restricted by $|z| < a$.

Consider a continuous Lyapunov function $V(t) > 0$, where $\forall t \in R^+$, $V(0)$ is bounded. It can be concluded that if the derivative of the Lyapunov function satisfies the following inequality, then $V(t)$ is bounded.⁽³¹⁾

$$\dot{V}(t) \leq -\rho V(t) + \sigma \quad (18)$$

Consider the Lyapunov function

$$V(t) = \sum_1^n \frac{a_i^2}{\pi} \tan\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) + \frac{1}{2} \mathbf{z}_2^T \mathbf{M}_x \mathbf{z}_2 + \frac{1}{2} \tilde{\mathbf{W}}_M^T \tilde{\mathbf{W}}_M + \frac{1}{2} \tilde{\mathbf{W}}_c^T \tilde{\mathbf{W}}_c + \frac{1}{2} \tilde{\mathbf{W}}_G^T \tilde{\mathbf{W}}_G, \quad (19)$$

$$\dot{V}(t) = \sum_1^n z_{1i} \dot{z}_{1i} \sec\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) + \frac{1}{2} \mathbf{z}_2^T \dot{\mathbf{M}}_x \mathbf{z}_2 + \mathbf{z}_2^T \mathbf{M}_x \dot{\mathbf{z}}_2 + \tilde{\mathbf{W}}_M^T \dot{\tilde{\mathbf{W}}}_M + \tilde{\mathbf{W}}_c^T \dot{\tilde{\mathbf{W}}}_c + \tilde{\mathbf{W}}_G^T \dot{\tilde{\mathbf{W}}}_G. \quad (20)$$

Because $2\mathbf{C}_x(\mathbf{q}, \dot{\mathbf{q}}) - \dot{\mathbf{M}}_x(\mathbf{q})$ is a skew-symmetric matrix, upon substituting Eqs. (13)–(16) into Eq. (20), we obtain

$$\begin{aligned} \dot{V}(t) = & -\sum_1^n k_{1i} z_{1i}^2 - z_2^T k_p z_2 - \sum_1^n \frac{k_{1i} a_i^2}{\pi} \tan\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) + z_2^T (\tilde{M}_x \dot{\alpha} + \tilde{C}_x \alpha + \tilde{G}_x) \\ & + z_2^T (\varepsilon_M \dot{\alpha} + \varepsilon_C \alpha + \varepsilon_G - k_r \operatorname{sgn}(z_2)) + \tilde{W}_M^T \dot{\hat{W}}_M + \tilde{W}_C^T \dot{\hat{W}}_C + \tilde{W}_G^T \dot{\hat{W}}_G. \end{aligned} \quad (21)$$

The stable adaptive update law of the RBFNN is

$$\dot{\hat{W}}_M = -S_M \dot{\alpha} z_2 - \beta_M \hat{W}_M, \quad (22)$$

$$\dot{\hat{W}}_C = -S_C \alpha z_2 - \beta_C \hat{W}_C, \quad (23)$$

$$\dot{\hat{W}}_G = -S_G z_2 - \beta_G \hat{W}_G, \quad (24)$$

where β_M, β_C , and β_G represent the robust items of the RBFNN. Substituting Eqs. (22)–(24) into Eq. (21) gives

$$\dot{V}(t) = -\sum_1^n k_{1i} z_{1i}^2 - z_2^T k_p z_2 - \sum_1^n \frac{k_{1i} a_i^2}{\pi} \tan\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) - \beta_M \tilde{W}_M^T \hat{W}_M - \beta_C \tilde{W}_C^T \hat{W}_C - \beta_G \tilde{W}_G^T \hat{W}_G \quad (25)$$

Taking the equality $-\tilde{W}^T W^* = -\tilde{W}^T (\tilde{W} + W^*)$ and Young's inequality $-\tilde{W}^T W^* \leq \frac{1}{2}(\tilde{W}^T W + W^{*T} W^*)$ into account, we obtain

$$\begin{aligned} \dot{V}(t) \leq & -z_2^T k_p z_2 - \sum_1^n \frac{k_{1i} a_i^2}{\pi} \tan\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) - \beta_M \tilde{W}_M^T \hat{W}_M - \beta_C \tilde{W}_C^T \hat{W}_C - \beta_G \tilde{W}_G^T \hat{W}_G \\ \leq & -\sum_1^n \frac{k_{1i} a_i^2}{\pi} \tan\left(\frac{\pi z_{1i}^2}{2a_i^2}\right) - z_2^T k_p z_2 - \frac{1}{2}(\tilde{W}_M^T \tilde{W}_M + \tilde{W}_C^T \tilde{W}_C + \tilde{W}_G^T \tilde{W}_G) \\ & + \frac{1}{2}(W_M^{*T} W_M^* + W_C^{*T} W_C^* + W_G^{*T} W_G^*). \end{aligned} \quad (26)$$

Equations (18) and (26) have the same form, from which we obtain

$$\begin{aligned} \sigma &= \frac{1}{2}(W_M^{*T} W_M^* + W_C^{*T} W_C^* + W_G^{*T} W_G^*), \\ \rho &= \min\left(2k_{11}, \dots, 2k_{1n}, \left(\frac{2\lambda_{\min}(k_p)}{\lambda_{\max} M_x}\right)\right). \end{aligned} \quad (27)$$

4. Experiments

Experiments are carried out with the Baxter robot manipulator, as shown in Fig. 4. The human partner interacts with the robot with his left arm. The robot recognizes his intention by identifying the gesture posed by his right hand. There is a ball held between the robot end effector and the left hand of the human partner. The end of the Baxter robot arm is equipped with a Mini45 force sensor to measure the contact force in real time. The human partner continuously adjusts the force he applies to the contact force exerted by the robot to keep the ball in place without falling off. Raw sEMG signals are collected by a Myo armband worn by the human partner on his right arm. In the first experiment, we verified that the proposed control framework could guarantee transient performance during interaction. In the second experiment, we verified that the proposed method could make the robot change its interaction mode and level automatically according to human intentions conveyed by hand gestures.

4.1 Test of RBFNN controller based on BLF

In the first experiment, we use a held balloon to interact with the left arm of the Baxter robot while ensuring transient performance. The initial Cartesian position is $[x, y, z] = [0.60, -0.33, 0.50]$ m and the final Cartesian position is $[x, y, z] = [0.60, 0.30, 0.50]$ m. We push the balloon along the Y-axis to make the robot's left arm move from the initial position to the final position and observe the variation of the force, position, and position error. Transient and steady performances are guaranteed by utilizing the BLF-based RBFNN constraint controller designed using Eq. (16). The parameter is $a_i = 0.04$ and the control gains are $k_{1i} = 80$, $k_p = 60$, and $k_r = 5$. Furthermore, to use the RBFNN to estimate the uncertain parameters of the robot dynamics, $n = 3^7$ neural network nodes are employed for $\hat{M}_x(q)$ and $\hat{G}_x(q)$ and $2n = 2 \times 3$ are employed for $\hat{C}_x(q)$. The

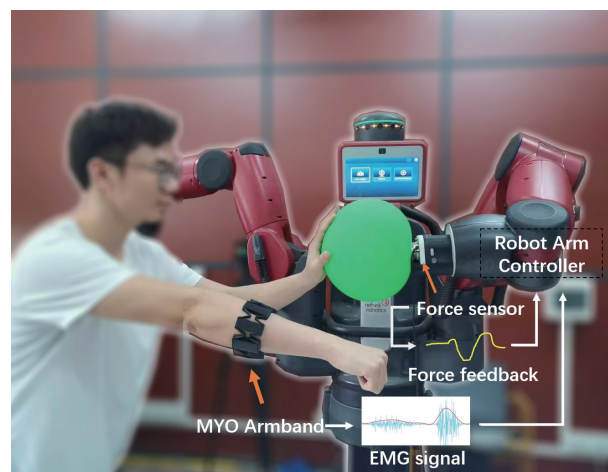


Fig. 4. (Color online) Process of HRI (left arm interacts with robot while right arm's hand gestures are used for human intention recognition).

weight matrices are initialized as $\hat{M}_x(\mathbf{0}) = \mathbf{0} \in \mathbb{R}^{n \times 1}$, $\hat{C}_x(\mathbf{0}) = \mathbf{0} \in \mathbb{R}^{2n \times 1}$, and $G_x(\mathbf{0}) = \mathbf{0} \in \mathbb{R}^{n \times 1}$, and the parameters of the RBFNN update law are chosen as $\beta_M = \beta_C = \beta_G = 5$.

The force on the Baxter robot's left arm during the interaction is shown in Fig. 5. The initial received force is not 0. This is because the Mini45 force sensor has errors due to gravity. The force received by the robot remains at about 3 N with tiny fluctuations. Because of ergonomic limitations, a human cannot exert an absolute constant force during interaction. The experimental results show that the admittance control is effective in force control for interaction with a flexible object.

The position and tracking error in the Y -axis are shown in Fig. 6. It can be seen from Fig. 6 that except for the larger tracking error at the beginning and end, the errors at other times are all approximately zero. The actual trajectory has good tracking performance for the reference trajectory, and the tracking error is kept within 4 cm, which is a small range. The experimental results show that our proposed BLF-based RBFNN controller limits the tracking error to a value within a certain range and ensures a transient response.

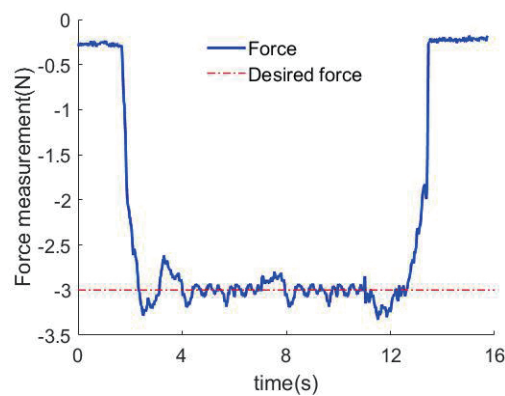


Fig. 5. (Color online) Force measured during HRI.

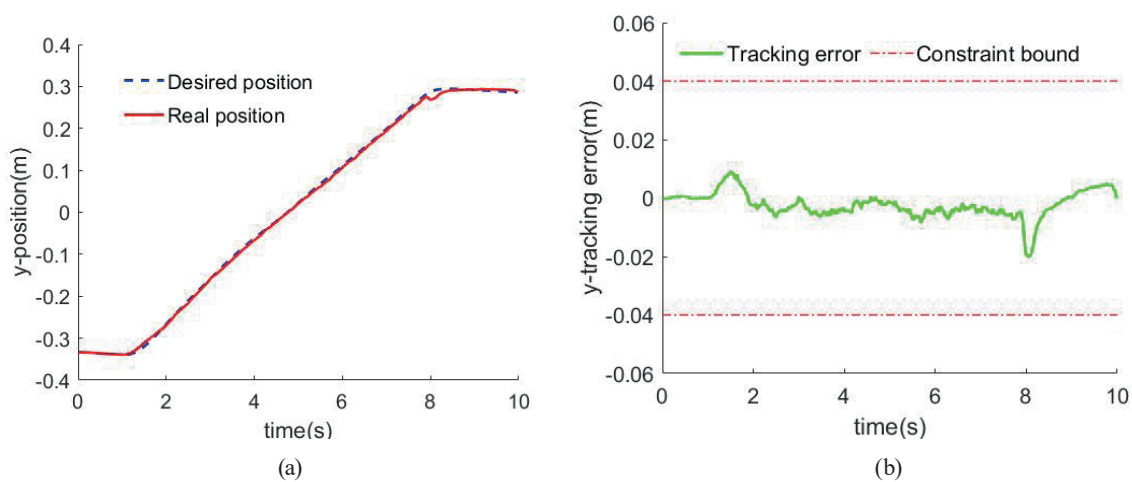


Fig. 6. (Color online) Y -axis position and tracking error during HRI. (a) Y -axis position and (b) Y -axis tracking error.

4.2 Human intention recognition based on sEMG signal

In the second experiment, we change the hand gestures to make the robot perform different interaction modes and levels to achieve human intention recognition. The raw sEMG signals corresponding to different gestures of a volunteer are shown in Fig. 7. It is observed that different features are associated with the hand gestures. The gestures are dynamic since static gestures require upper limb muscles to stay taut all the time, which will inevitably produce muscle fatigue and lower the success rate of gesture recognition. The recognition learning curves of different gestures are shown in Fig. 8. As the number of iterations increases, the loss decreases and the accuracy increases gradually, indicating that the features of different hand gestures could be acquired after passing through a CNN classifier to achieve good classification performance.

The robot could change its interaction mode and level according to human intentions, as described in detail in Sect. 3.1. According to the classification result of the hand gestures, we first set *flag1* to choose the interaction mode. Then, by recognizing the next hand gesture, we changed the value of *flag2* to define the interaction level. Figure 9 shows the variation of the interaction force for three sample conditions. Specifically, Fig. 9(a) shows the force switching to a different level in the same mode. The red line represents the force in the forward mode and in level 1, and the green line represents the force in the forward mode and in level 2, which indicates an increased force level. Figure 9(b) shows the force switching to a different mode in the same level. The red line represents the force in the forward mode and in level 1, and the

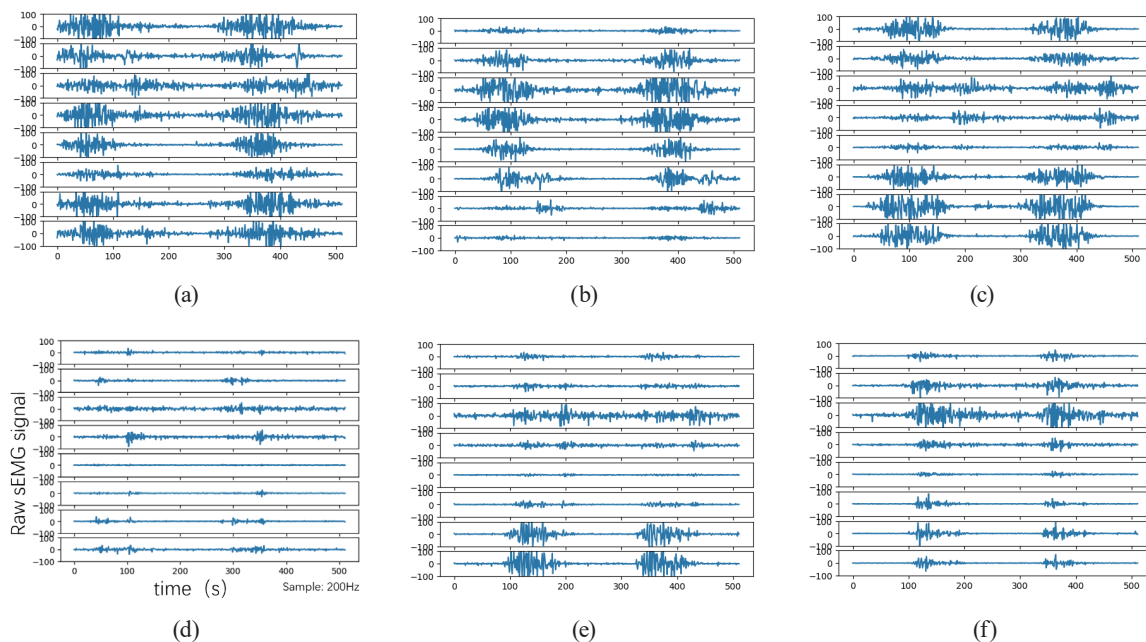


Fig. 7. (Color online) Raw sEMG signals corresponding to different gestures: (a) gesture 1, (b) gesture 2, (c) gesture 3, (d) gesture 4, (e) gesture 5, and (f) gesture 6.

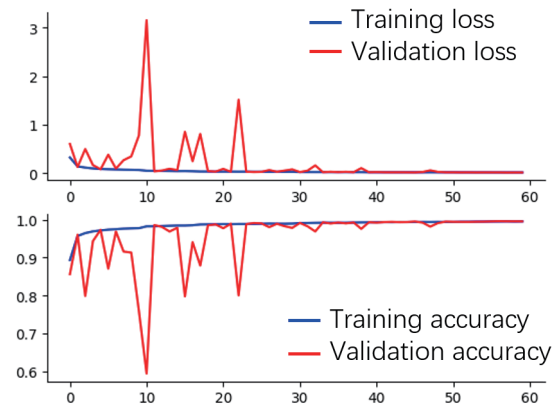


Fig. 8. (Color online) Recognition learning curves of different gestures.

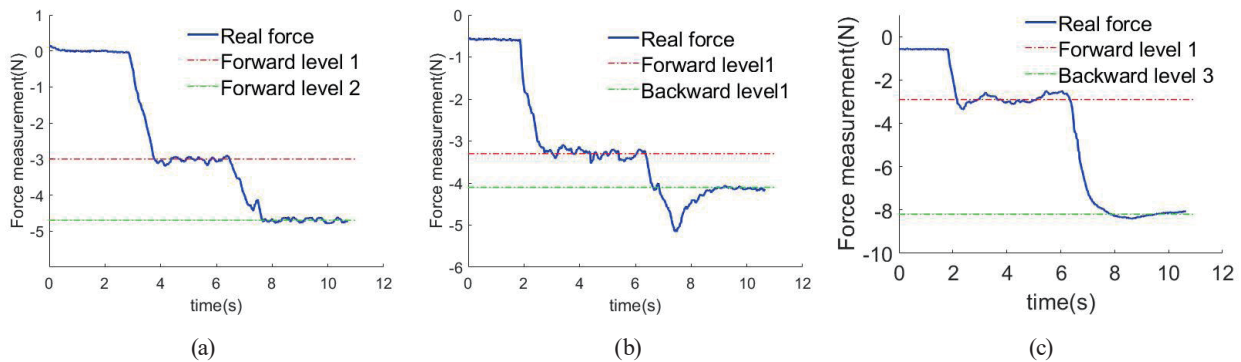


Fig. 9. (Color online) Force measurement in different interaction modes and levels. (a) Switching to different level in same mode. (b) Switching to different mode in same level. (c) Switching to different mode and level.

green line represents the force in the backward mode and in level 1. Because a virtual force is added according to the different level in the backward mode with level 1 for both interaction levels, the absolute values of the forces are different in Fig. 9(b) for the forward and backward modes. Moreover, although the forces are both in the negative direction, the direction of motion is opposite for these two modes. Owing to the accuracy of the sensor, the forces measured in the forward mode and in level 1 in Figs. 9(a) and 9(b) are not exactly the same. However, the variation range is within 0.5 N. Figure 9(c) shows the force switching to a different mode and a different level. The red line represents the force in the forward mode and in level 1, and the green line represents the force in the backward mode and in level 3.

The above experimental results prove that our proposed method can effectively identify human intentions and flexibly achieve intelligent HRI. Humans can change the interaction mode and level according to different situations when interacting with a robot while the contact force fluctuates around a certain value.

5. Conclusions

In this paper, we propose an algorithm of human intention recognition through identifying various hand gestures, which makes the robot automatically change the interaction scheme to adapt to different HRI tasks. An admittance control framework combining a tan-type BLF and an RBFNN has also been proposed to achieve desirable interaction and tracking performance while guaranteeing stability. HRI experiments with a Baxter robot show that, first, the proposed control framework gives the robot good tracking and transient performances. Second, the human intention recognition algorithm allows the robot to adapt its interaction mode and level according to human hand gestures in real time, which ensures the exertion of the most appropriate contact forces by the robot, greatly improving the success rate of the HRI task. This framework will be extended to accommodate more challenging scenarios and tasks in the future by exploring the relationship between the activity level of human upper limb muscles and the robot position in an HRI process to achieve continuous intention recognition, and by realizing optimal interaction with flexible objects.

Acknowledgments

This work was supported in part by the National Nature Science Foundation of China (NSFC) under Grant 61803039.

References

- 1 W. Wan, F. Lu, Z. Wu, and K. Harada: *Neurocomputing* **259** (2017) 85. <https://doi.org/10.1016/j.neucom.2017.01.077>
- 2 W. Wan, K. Harada, and F. Kanehiro: *IEEE Trans. Ind. Inf.* **16** (2019) 442. <https://doi.org/10.1109/tii.2019.2892772>
- 3 L. Randazzo, I. Iturrate, S. Perdakis, and J. D. R. Millán: *IEEE Rob. Autom. Lett.* **3** (2017) 500. <https://doi.org/10.1109/lra.2017.2771329>
- 4 H. Su, Y. Hu, H. R. Karimi, A. Knoll, G. Ferrigno, and E. D. Momi: *Neural Networks* **131** (2020) 291. <https://doi.org/10.1016/j.neunet.2020.07.033>
- 5 S. Murata, Y. Li, H. Arie, T. Ogata, and S. Sugano: *IEEE Trans. Cognit. Dev. Syst.* **10** (2018) 712. <https://doi.org/10.1109/tcds.2018.2797260>
- 6 M. H. Raibert and J. J. Craig: *J. Dyn. Syst. Meas. Contr.* **102** (1981) 127. <https://doi.org/10.1115/1.3139652>
- 7 M. C. Yip and D. B. Camarillo: *IEEE Rob. Autom. Lett.* **1** (2016) 844. <https://doi.org/10.1109/lra.2016.2526062>
- 8 Y. Jiang, C. Yang, Y. Wang, Z. Ju, Y. Li, and C. Y. Su: *Mechatronics* **67** (2020) 102348. <https://doi.org/10.1016/j.mechatronics.2020.102348>
- 9 C. Ott, R. Mukherjee, and Y. Nakamura: *J. Intell. Rob. Syst.* **78** (2015) 359. <https://doi.org/10.1007/s10846-014-0082-1>
- 10 G. Peng, C. Yang, W. He, and C. L. P. Chen: *IEEE Trans. Ind. Electron.* **67** (2019) 3138. <https://doi.org/10.1109/tie.2019.2912781>
- 11 C. Yang, G. Peng, Y. Li, R. Cui, L. Cheng, and Z. Li: *IEEE Trans. Cybern.* **49** (2018) 2568. <https://doi.org/10.1109/tcyb.2018.2828654>
- 12 C. Yang, G. Peng, L. Cheng, J. Na, and Z. Li: *IEEE Trans. Syst. Man Cybern.: Syst.* **51** (2019) 1. <https://doi.org/10.1109/tsmc.2019.2920870>
- 13 Y. Jiang, Y. Wang, Z. Miao, J. Na, Z. Zhao, and C. Yang: *IEEE Trans. Neural Networks Learn. Syst.* (2020). <https://doi.org/10.1109/tnnls.2020.3037795>
- 14 C. Yang, C. Chen, W. He, R. Cui, and Z. Li: *IEEE Trans. Neural Networks Learn. Syst.* **30** (2018) 777. <https://doi.org/10.1109/tnnls.2018.2852711>

- 15 D. Y. Li, P. Li, W. C. Cai, Y. D. Song, and H. J. Chen: IEEE Trans. Ind. Electron. **65** (2017) 3275. <https://doi.org/10.1109/tie.2017.2748036>
- 16 C. Yang, Y. Jiang, Z. Li, W. He, and C. Y. Su: IEEE Trans. Ind. Inf. **13** (2016) 1162. <https://doi.org/10.1109/tii.2016.2612646>
- 17 W. Sun, Z. Zhu, J. Lan, and Y. Peng: Trans. Inst. Meas. Control **42** (2020) 1180. <https://doi.org/10.1177/0142331219886411>
- 18 C. Yang, D. Huang, W. He, and L. Cheng: IEEE Trans. Neural Networks Learn. Syst. (2020) 1. <https://doi.org/10.1109/tnnls.2020.3017202>
- 19 Z. Fang and A. M. López: IEEE Trans. Intell. Transp. Syst. **21** (2019) 4773. <https://doi.org/10.1109/tits.2019.2946642>
- 20 D. Lee and Y. Park: IEEE Trans. Consum. Electron. **55** (2009) 2308. <https://doi.org/10.1109/tce.2009.5373803>
- 21 K. Zinchenko, C. Y. Wu, and K. T. Song: IEEE Trans. Ind. Inf. **13** (2016) 607. <https://doi.org/10.1109/tii.2016.2625818>
- 22 E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh: arXiv preprint arXiv:19 (2019) 2153. <https://doi.org/10.18653/v1/d19-1221>
- 23 C. Yang, H. Wu, Z. Li, W. He, N. Wang, and C. Y. Su: IEEE Trans. Ind. Inf. **14** (2017) 3822. <https://doi.org/10.1109/tii.2017.2785415>
- 24 C. Zeng, C. Yang, Z. Chen, and S. L. Dai: Assembly Automation (2018). <https://doi.org/10.1108/aa-02-2018-019>
- 25 H. Kong, C. Yang, G. Li, and S. L. Dai: IEEE Access **8** (2020) 26030. <https://doi.org/10.1109/access.2020.2970468>
- 26 C. Zeng, C. Yang, H. Cheng, Y. Li, and S. L. Dai: IEEE Trans. Ind. Inf. **17** (2020) 1244. <https://doi.org/10.1109/tii.2020.2984482>
- 27 C. Yang, J. Luo, C. Liu, M. Li, and S. L. Dai: IEEE Trans. Autom. Sci. Eng. **16** (2018) 1512. <https://doi.org/10.1109/tase.2018.2874454>
- 28 R. M. Murray, Z. Li, and S. S. Sastry: A Mathematical Introduction to Robotic Manipulation (CRC press, Florida, 1994) Chap. 2. <https://doi.org/10.1201/9781315136370>
- 29 T. H. Lee and C. J. Harris: Adaptive Neural Network Control of Robotic Manipulators (World Scientific, Singapore, 1998) p. 396. <https://doi.org/10.1142/3774>
- 30 X. Jin and J. X. Xu: Automatica **49** (2013) 2508. <https://doi.org/10.1016/j.automatica.2013.04.039>
- 31 H. Huang, C. Yang, and C. P. Chen: IEEE Trans. Cybern. (2020) 1. <https://doi.org/10.1109/tcyb.2020.2998984>

About the Authors



Junbao Gan received his B.S. degree in automation from South China University of Technology, Guangzhou, China, in 2018, where he is currently pursuing his M.S. degree. His current research interests include human–robot interaction, human intention recognition, and adaptive control.



Ning Wang is a senior lecturer in robotics at the Bristol Robotics Laboratory, University of the West of England, United Kingdom. She received her M.Phil. and Ph.D. degrees in electronics engineering from the Department of Electronics Engineering, The Chinese University of Hong Kong, Hong Kong, in 2007 and 2011, respectively. Ning has rich project experience: she has been a key member of the EU FP7 Project ROBOT-ERA, the EU Regional Development Funded Project ASTUTE 2020, and industrial projects with UK companies. She has received several awards including Best Paper Award of ICIRA'15, Best Student Paper Award nomination of ISCSLP'10, and Award of

Merit of the 2008 IEEE Signal Processing Postgraduate Forum. Her research interests lie in signal processing, intelligent data analysis, human–robot interaction, and autonomous driving.



Lei Zuo (M'17) received his B.Sc. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 2011 and 2017, respectively. Since 2017, he has been with the School of Electronic and Control Engineering, Chang'an University, Xi'an, where he is currently an assistant professor. His current research interests include multiagent systems, coverage control, model predictive control, and vehicle platoon control.