

Received August 20, 2019, accepted September 12, 2019, date of publication September 20, 2019, date of current version October 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2942648

Development of a Neural Network-Based Control System for the DLR-HIT II Robot Hand Using Leap Motion

CHUNXU LI^{1,2}, ASHRAF FAHMY^{1,3}, AND JOHANN SIENZ¹

¹Advanced Sustainable Manufacturing Technologies 2020, Swansea University, Swansea SA1 8EN, U.K.

²School of Automation and Electrical Engineering, Qingdao University of Science and Technology, Qingdao 266042, China

³Department of Electrical Power and Machines, Helwan University, Helwan 11795, Egypt

Corresponding author: Chunxu Li (chunxu.li@swansea.ac.uk)

ABSTRACT In this paper, a neural network (NN) based adaptive controller has been successfully developed for the teleoperation of a DLR-HIT II robot hand using Leap Motion sensor. To achieve this, the coordinate positions of the human hand fingers are captured by a Leap Motion sensor. Moreover, inverse kinematics is used to transform the Cartesian position data of the fingertips into corresponding joint angles of all the five fingers, which are then directly sent to teleoperate the DLR-HIT II hand via User Datagram Protocol (UDP). In addition, a NN-based control system programmed by the MATLAB Simulink function has been investigated to enhance the overall control performance by compensating the dynamic uncertainties existing in the teleoperation system. The stability of the control system has been proved mathematically using Lyapunov stability equations. A series of experiments have been conducted to test the performance of the proposed control technique, which has been proved to be an effective teleoperation strategy for the DLR-HIT II robot hand.

INDEX TERMS Neural networks, teleoperation, DLR-HIT II robot hand, leap motion, inverse kinematics.

I. INTRODUCTION

In recent years, there has been a rapid advancement in robotics, with its applications growing extensively in numerous fields. Development in robotics has greatly facilitated people's lives and is one of the most rapidly emerging industries of today's daily life and manufacturing usage. Most countries are paying much attention to the development of robots, such as the National Robot Program of the United States, the Horizon 2020 of the European Union and the National Natural Science Foundation of China's 863 Programme [1]. From the beginning of the Industrial 4.0 revolution, the role of robots has become increasingly significant with the advance of science and technology. However, this role remains largely limited to traditional industrial applications such as pre-set and repeated tasks. At present, it remains a challenge to develop robots that are capable of performing an assortment of tasks in unknown, complex and dynamic environments [2]. Besides, service-oriented, medical as well

as industrial robots are required to work in complex external environments. Hence, control of robots in unknown and dynamic environments has been widely studied and a higher demand for advanced intelligence in robots has been recognized. It is noteworthy that in the working environment, robots are inevitably affected by unpredictable and uncertain external disturbances. In this regard, traditional control methods are not sufficient for qualifying them in terms of intellectuality. Recent research has been actively investigated into the interaction and control between robots and external forces. Interactive control technology cannot only assure compliance of robot behaviour, but also provide a relatively safe operating environment for operators, which are impossible to achieve through traditional methods of robot control.

According to [3], the human-robot interaction (HRI) is a sub-area of the human-computer interaction (HCI), which studies the interaction between humans and robots, and focuses on developing more intelligent and anthropomorphic robots. HRI is widely used in research and implementation of robot systems in hazardous zones where human involvement needs to be minimized and remote operation

The associate editor coordinating the review of this manuscript and approving it for publication was Ming Luo.



FIGURE 1. The DLR-HIT II dexterous robot hand.

of robots is required. These inventions can also apply to care for the elderly and the disabled personnel, as well as to entertainment purposes [4]. There are various methods of human-robot interaction, amongst which, two are considered the main methods of — physical interaction and teleoperation interaction. For specialized tasks, such as medical operations, the safest method for human-robot interaction is teleoperation [5]. In this method, the user interfaces with the robot through a haptic signal for coordinated feedback. The research [6] carried out experiments to characterize this problem, and proposed several methods to provide haptic feedback in order to improve a surgeon's performance. However, they do not take the enough required control room for the surgeon into consideration. As already known, the surgeon's activities have high requirements for cleaning, and thus the remote control is a better choice. In the research [7], a research team introduces a visual impedance scheme for vision-based control of the robot so as to realize task-level dynamical control. In the control scheme, the image features are applied to the impedance equation so that integration of a visual servo and a conventional servo system could be accomplished.

The interaction between robots and complex environments can lead to potential instability, inaccuracy, mechanical non-linearity and saturation, thus may have a negative impact on the overall working system. At present, there are two main methods of interactive control between robot and environment external forces namely, the hybrid force/position control and impedance/admittance control [8]. Hybrid force/position control has the potential of achieving good control performance and anti-disturbance ability. However, in case of a

strong interaction between the robot and the environment, the hybrid force/position control method often leads to system instability and security problems [9]. As compared to hybrid force/position control, impedance/admittance control aims to establish the correct balance in the relationship between the robot and interactive force received from the external environment. Through impedance/admittance control method, robot's behaviour can be adapted to the environment and the overall stability of the control system can be ensured [10]. Therefore, further study and development of robot interactive control technology for dealing with various uncertainties in a real-world situation is necessary so as to ensure that robots can work in a dynamic and unknown environment. Hence, the development of robot interactive control technology is of great practical significance.

There are dynamic uncertainties in a robot's control system or during the modelling which have consequences in the learning and reproducing performance. These uncertainties can be controlled with the development of controls using Fuzzy Logic or Radial Basis Function (RBF) Networks. RBF is a three-layer forward network with a single hidden layer [11]. Its first layer is an input layer and consists of signal source nodes, and its second layer is a hidden calculation layer. The total number of hidden layer nodes depends on the complexity and requirements of the problem described. The transformation function of the neurons in the hidden layer, for example the radial basis function, is a non-negative linear function that is radially symmetric and attenuated to the center point [11]. RBF is a local-response function, and the specific local-response is reflected in the transformation of its visible layer to the hidden layer, a phenomenon which differs it from other networks. The third layer is the output layer which responds to the input mode, and therein the input layer only acts as a transmission signal layer [12]. Previous forward network transformation functions were used for a global response. Functionally, the input layer and the hidden layer can be connected with a weight factor. The output layer and the hidden layer perform different tasks, hence learning strategies used for them will also be different [13]. The output layer adjusts to the linear weight using a linear optimization strategy, so the learning speed becomes faster. However, the hidden layer adjusts parameters of the activation function (Green's function and Gaussian function, more commonly the latter) through nonlinear optimization strategies which make the learning process slower [14].

Recently, researches on the dexterous robot hands have been widely investigated. For shortening the distance of power transmission, the embedded actuator method has been used to integrate all the actuators inside the hand, fingers or joints. The Gifu Hand [15], the DLR Hand II [16] and the DLR-HIT Hand I [17] have taken this method for their robot hands products. According to [18], the Stanford-JPL Hand, the Utah/MIT Hand, the NASA Robonaut Hand and the DLR Hand Arm System are designed with a different method, where all the actuators are mounted outside the robot fingers and driven through tendon cables. In this paper, the modelling

and control schemes of the dexterous robot hand DLR-HIT II (shown in Fig.1) have been demonstrated, which is a multisensory five-finger robot hand with a total of 15 DOFs and more than 100 sensors produced by German Aerospace Center (DLR) and Harbin Institute of Technology (HIT).

Commonly, the robot hand alone is not widely studied and the research works only on a robot hand itself are limited comparing to those on the robot manipulators. Some research institutes and industries apply the robot hands to a robot manipulator for their studies and operations to do some specific tasks, such as the pick-and-place tasks with grasping and opening actions of the robot hands [19]. In this case, the robot hand is implemented as an overall object so that its control method normally only contributes to the clenching & unclenching of the robot hand. In addition, when a robot hand is used to grasp a heavy object, the gravity and inertia can cause problems to the control system. In this paper, a novelty robot hand tele-control system mixed with an adaptive function approximate method has been developed, where all the fingers can be separately controlled with overcoming of the unknown dynamics.

II. HAND GESTURE RECOGNITION USING LEAP MOTION

In this paper, the Leap Motion somatosensor (shown in Fig.2) produced by Leap Motion Inc., a US-based somatosensory controller manufacturer, released on February of 2013 was employed for tele-control purposes. It features a natural gesture function to control computer programs that replaces the need for traditional mouse and keyboard controls [20]. By acquiring user gestures in spatial motion data, we can naturally manipulate 3D objects using familiar gestures. The Leap Motion was a major breakthrough in the accuracy, efficiency, and portability aspects, where the accuracy of the Leap Motion is up to 200 times greater than other somatosensory products such as Microsoft's Kinect [21]. It has been applied in many human-computer interaction roles such as education, medicine, games, and Google Maps 3D due to its intuitive nature and depth of control that can allow for circumvention of the requirements for a mouse or touch screen.

Internally, a Leap Motion device consists of two narrow-band filters, two mid-low-pixel cameras, three infrared LEDs, and a USB high-speed chip in [22]. The principle operation of Leap Motion is to capture a hand image illuminated by the infrared LED lamp through two cameras and establish a three-dimensional model of the hand by analysing the change of gesture in the stereo camera images. When the hand enters the field of view of the device, the stereo camera system begins capturing the target and calculates the parallax of the target in real time so that its spatial information is obtained [20]. The built-in communication chip in the body uses standard USB transmission technology to digitise the collected image information, and transmit the converted data to the computer. The computer then restores the gesture changes through image recognition

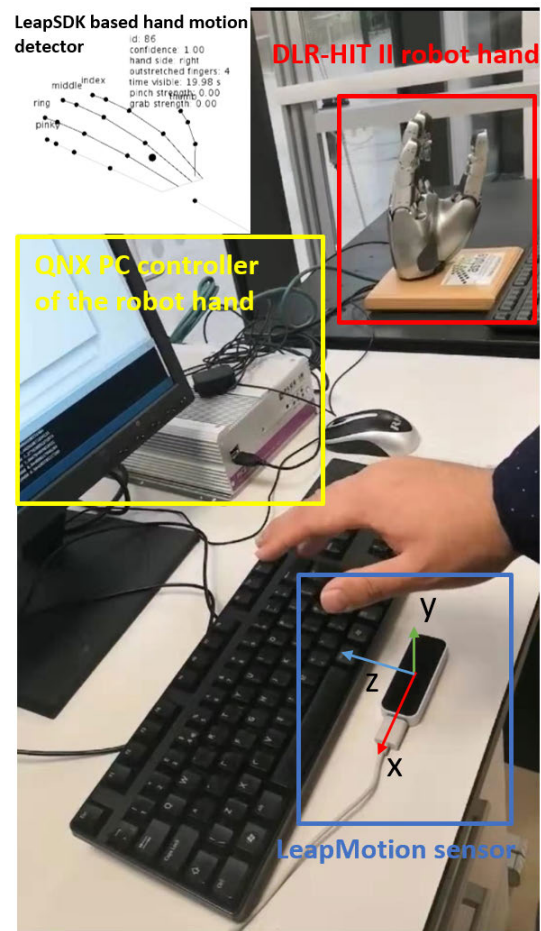


FIGURE 2. Illustration of the teleoperation system.

and calculation and resubmits hand motion to a PC application for the usages by the programmer.

Fig.2 above shows the system configuration of this research. Here the Leap Motion is used to collect the fingertips' data in Cartesian space and then those data are transmitted to corresponding angle values by applying the inverse kinematics calculations. In addition, a remote master PC connects to the DLR-HIT II robot hand's QNX controller by an Ethernet cable, where the NN-based adaptive controller is running on the Simulink of the MATLAB. The methodologies for our proposed teleoperation system are demonstrated in the following sections.

III. ACQUISITION OF FINGER JOINT ANGLES

A. HAND MODELLING

In general, the model of the human hand can be thought of as 5 mechanical rigid body models, which are representing 5 fingers respectively. The index finger, middle finger, ring finger and the ring finger all consist of a proximal interphalangeal (PIP) joint, a distal interphalangeal joint (DIP), and a metacarpophalangeal (MCP) joint. The thumb differs, only consists interphalangeal (IP) joint and MCP joint. In addition,

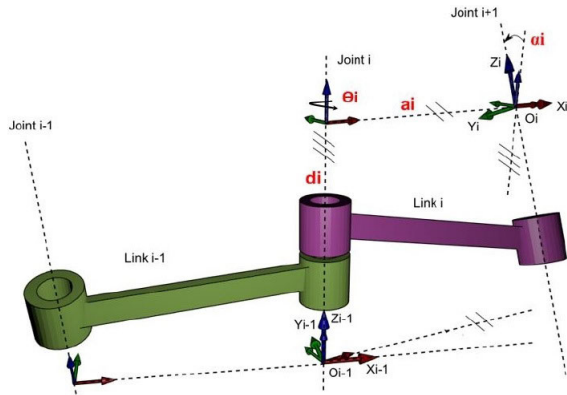


FIGURE 3. The DH model parameter and definition [24].

there are five carpometacarpal (CMC) joints in the palm. According to the characteristics of kinematics, the thumb is composed of three degrees of freedom (DoF), and the three joints are all rotating joints, while all the other fingers have 4 DoFs. This paper builds a model of the finger by establishing a Denavit-Hartenberg (DH) model of the human hand, which is a method proposed by Denavit and Hartenberg in 1955 to describe the relationship between the joints and the connecting rods of the robot arm using the geometry of the system [23]. This method fixes a coordinate system on each link of the modelled robot. The homogeneous transformation is used to obtain the spatial relationship between the adjacent two links. The sequential transformation can finally derive the pose of the end effector relative to the base coordinate system to establish the kinematic equation of the robot. This paper will present the method of building a robot model and construct a dynamic equation. Fig.3 shows the DH model representation and its associated parameter definitions.

θ_i indicates the angle between x_{i-1} and x_i along the z_{i-1} direction; d_i indicates the distance between x_{i-1} and x_i along the z_{i-1} direction; α_i indicates the angle between z_{i-1} and z_i along the x_i direction; a_i indicates the distance between z_{i-1} and z_i along the x_i direction. In general, d_i , a_i , and α_i are constants, and for a rotating joint, θ_i is a joint variable. According to the DH modeling method, combined with the structure of the human finger, the kinematic model of the human hand is given as shown in Fig.4.

There is no any motor in the robot hand palm so that we only need to take the robot hand fingers into account. In Fig.4, we use the “o” point as the center point of the first knuckle and establish a coordinate system. In the “o” coordinate system, the direction of the axis z is perpendicular to the palm plane upward, and the direction of the axis x is upward. According to the structural difference, here we define that there are two cases to be modelled, which are the thumb and the rest. L_i represents the distance between each finger joint. Taking the ring finger as an example of “rest”, the corresponding DH parameters for the two above mentioned cases are given in Tab.1 and Tab.2. Once the DH parameters tables are obtained, the model of the whole hand can be successfully established.

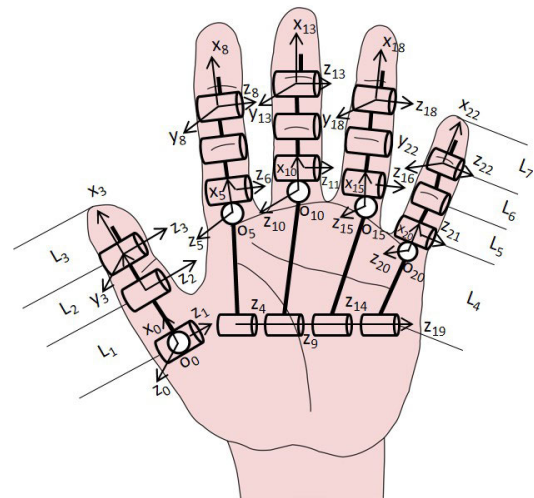


FIGURE 4. The DH model of the human hand.

TABLE 1. Model characters table of thumb using Denavit & Hartenberg’s method.

LinkNumber	θ_i	$d_i(m)$	$a_i(m)$	$\alpha_i(rad)$
1	θ_{CMC1}	0	0	$\pi/2$
2	θ_{MCP2}	0	L_1	0
3	θ_{MCP}	0	L_2	0
4	θ_{IP}	0	L_3	0

TABLE 2. Model characters table of “rest” using Denavit & Hartenberg’s method.

LinkNumber	θ_i	$d_i(m)$	$a_i(m)$	$\alpha_i(rad)$
1	θ_{CMC}	0	L_4	$\pi/2$
2	θ_{MCP1}	0	0	$-\pi/2$
3	θ_{MCP2}	0	L_5	0
4	θ_{PIP}	0	L_6	0
5	θ_{DIP}	0	L_7	0

B. VECTOR APPROACH BASED INVERSE KINEMATICS

The vector approach based inverse kinematics is used to calculate the joint angles from the end effector’s coordinate location, which can be obtained directly from the Leap Motion SDK. It returns the position of every single fingertip in Cartesian space, which can be converted to the required joint angles of fingers ($\theta_1 - \theta_4$), which will be used for the neural networks aspects later. The distance between two points in the 3D space can be expressed as:

$$d_{3D} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (1)$$

where (x_1, y_1, z_1) and (x_2, y_2, z_2) are points in the 3D space, d_{3D} is the length between these two points. Using the angle formula of the space vector:

$$\cos \langle \vec{a}, \vec{b} \rangle = \frac{\vec{a} \cdot \vec{b}}{|\vec{a}| \cdot |\vec{b}|} \quad (2)$$

where the $\vec{a} \cdot \vec{b}$ can be obtained by giving the spatial coordinate information of Cartesian coordinates of four degrees of freedom of the three joints of the human finger returned by the Leap Motion SDKs; $|\vec{a}| \cdot |\vec{b}|$ can be calculated by equation (1). Thus, all the k fingers' joints angles can be obtained using the vector approach based inverse kinematics methodology.

IV. NEURAL NETWORKS BASED CONTROL SYSTEM

In adaptive control, a neural network is a commonly used function approximation tool. The unknown function term in the control system is estimated by the neural network. A neural network can be grouped into linear parameterization and nonlinear parameterization, corresponding to RBFNNs and multi-layer neural network (MNNs). RBFNN not only is simple in structure and fast in learning, but also avoids the problems of many nerve layers. As a result, the RBFNN meets the real-time requirements of the control system. Therefore, in this paper, the RBFNN is used to approximate the unknown terms of the control system.

In control engineering, the universal approximation property of RBF neural network has been widely used as a function approximator. RBF neural network consists of an input layer, a hidden layer and an output layer. The input layer to the hidden layer is a nonlinear transformation, and the output layer is a linear combination of the output of the hidden layer. Therefore, the RBF neural network belongs to the linear parameterized network. In this paper, a continuous function can be approximated using linear parameter RBF neural networks, such as $F(z) : R^m \rightarrow R$, over a minimized set $\Omega_z \subset R^m$. This can be formulated as, in [25]:

$$F(z) = W^{*T}S(z) + \epsilon_z \quad \forall z \in \Omega_z \quad (3)$$

where $W^* = [w_1^*, w_2^*, \dots, w_l^*]^T \in R^l$ is the weight vector, $z \in \Omega_z$ is the input vector with $\Omega_z \subset R^m$ being a minimal set, l is the NN node number, and ϵ_z is the estimation error. $S(z) = [S_1(\|z - \mu_1\|), \dots, S_l(\|z - \mu_l\|)]^T$, is the regressor vector, with a radial basis function $S_i(\cdot)$, and μ_i ($i = 1, \dots, l$) a central inside $S_i(\cdot)$. The Gaussian functions are presented as follows:

$$S_i(\|z - \mu_i\|) = \exp \left[\frac{-(z - \mu_i)^T(z - \mu_i)}{\varsigma^2} \right] \quad (4)$$

where ς is the variance and $\mu_i = [\mu_{i1}, \mu_{i2}, \dots, \mu_{im}]^T \in R^m$ represents the center of each receptive area. The notation of the data output from the result of the inverse kinematics, in addition to the desired joint space trajectory of the i_{th} finger q_{di} is defined as:

$$q_{di} = [q_{di1}, q_{di2}, q_{di3}, q_{di4}]^T \in R^4. \quad (5)$$

Here, applying the NN control procedure to achieve the accurate tracking of the reference joint trajectory is used. Hence, the controller should be fast respond, stable and accurate. The reference joint trajectory of the i_{th} finger of the robot hand $q_{di} \in R^4$ consists of a time series of the angle values that are generated from the motion data obtained by the

inverse kinematics. The angular matrix $q_i \in R^4$ denotes the actual joint positions of the i_{th} finger during the teleoperation. At that point, as indicated by [25], the dynamics of the robot hand's i_{th} finger are given as (6).

$$\tau_i = M_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i)\dot{q}_i + G_i(q_i) + U_i(q_i) \quad (6)$$

where $M(q) \in R^{4 \times 4}$ is the inertia matrix, $C(q, \dot{q}) \in R^{4 \times 4}$ is the Coriolis matrix, $G(q) \in R^{4 \times 1}$ are the gravity terms and $U(q) \in R^{4 \times 1}$ are unmodeled elements owing to the system uncertainties.

The dynamics of all the k fingers can be defined in q as follows [18]:

$$\tau = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + U(q) \quad (7)$$

where $M(q) = \text{diag}\{M_1(q_1), \dots, M_k(q_k)\}$, $C(q, \dot{q}) = \text{diag}\{C_1(q_1, \dot{q}_1), \dots, C_k(q_k, \dot{q}_k)\}$, $G(q) = [G_1^T q(1), \dots, G_k^T q(k)]$, $U(q) = [U_1^T q(1), \dots, U_k^T q(k)]$ and $\tau = [\tau_1^T, \dots, \tau_k^T]^T$ [26].

Define $z = \dot{e}_q + \Lambda e_q$, $q_r = \dot{q}_d - \Lambda e_q$, where $e_q = q - q_d$, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, λ_n is a positive constant. Here the dynamic equation (7) can be rewritten as (8) from [25].

$$\tau - U(q) = M(q)\dot{z} + C(q, \dot{q})z + G(q) + M(q)\dot{q}_r + C(q, \dot{q})q_r \quad (8)$$

In [25], an adaptive controller is designed as (9),

$$\tau = \hat{H}(q) + \hat{M}(q)\dot{q}_r + \hat{C}(q, \dot{q})q_r - Kz \quad (9)$$

where $K = \text{diag}(k_1, k_2, \dots, k_i)$, $k_i > \frac{1}{2}$, $\hat{H}(q)$, $\hat{M}(q)$ and $\hat{C}(q, \dot{q})$ are the RBFNN estimates of $G(q) + U(q)$, $M(q)$ and $C(q, \dot{q})$ respectively.

Then the closed circling framework dynamic equation is given below as (10) [25].

$$M\dot{z} + Cz + Kz = (\hat{M} - M)\dot{q}_r + (\hat{C} - C)q_r + (\hat{H} - H) \quad (10)$$

The RBFNN based approximation approach is applied as follows [25].

$$\begin{aligned} M(q) &= W_M^{*T}S_M(q) + \epsilon_M \\ C(q, \dot{q}) &= W_C^{*T}S_C(q, \dot{q}) + \epsilon_C \\ H(q) &= W_H^{*T}S_H(q) + \epsilon_H \end{aligned} \quad (11)$$

where W_M^* , W_C^* and W_H^* are the constant ideal weight matrix; $S_M(q)$, $S_C(q, \dot{q})$ and $S_H(q)$ are the basis function matrix, and ϵ_M , ϵ_C and ϵ_H are the mismatch uncertainties due to the fact that the number of the hidden neuron is limited.

Then the equation for basis function matrix can be written as follow [25],

$$\begin{aligned} S_M(q) &= \text{diag}(S_q, \dots, S_q) \\ S_C(q, \dot{q}) &= \text{diag} \left(\begin{bmatrix} S_q \\ S_{\dot{q}} \end{bmatrix}, \dots, \begin{bmatrix} S_q \\ S_{\dot{q}} \end{bmatrix} \right) \\ S_H(q) &= [S_q^T \dots S_q^T]^T \end{aligned} \quad (12)$$

where

$$\begin{aligned} S_q &= [\phi(\|q - q_1\|) \ \phi(\|q - q_2\|) \ \dots \ \phi(\|q - q_n\|)]^T \\ S_{\dot{q}} &= [\phi(\|\dot{q} - \dot{q}_1\|) \ \phi(\|\dot{q} - \dot{q}_2\|) \ \dots \ \phi(\|\dot{q} - \dot{q}_n\|)]^T \end{aligned} \quad (13)$$

The estimates of $M(q)$, $C(q, \dot{q})$ and $H(q)$ can be obtained as (14).

$$\begin{aligned} \hat{M}(q) &= \hat{W}_M^T S_M(q) \\ \hat{C}(q, \dot{q}) &= \hat{W}_C^T S_C(q, \dot{q}) \\ \hat{H}(q) &= \hat{W}_H^T S_H(q) \end{aligned} \quad (14)$$

Now, substituting (14) into (10), then the previous equations are simplified as

$$\begin{aligned} M\dot{z} + Cz + Kz &= \tilde{W}_M^T S_M(q)\dot{q}_r \\ &\quad + \tilde{W}_C^T S_C(q, \dot{q})q_r + \tilde{W}_H^T S_H(q) \end{aligned} \quad (15)$$

where $\tilde{W}_M^T = \hat{W}_M^T - W_M^{*T}$, $\tilde{W}_C^T = \hat{W}_C^T - W_C^{*T}$ and $\tilde{W}_H^T = \hat{W}_H^T - W_H^{*T}$.

For a matrix of $n \times n$, the sum of the elements on the main diagonal of the matrix (the diagonal from the upper left to the lower right) is called the trace of matrix A, denoted $tr(A)$; if A, B are $m \times n$ and $n \times m$ matrices, then $tr(AB) = tr(BA)$ [27].

V. STABILITY GUARANTEE OF CONTROL SYSTEM

Here, the Lyapunov candidate function is chosen as,

$$\begin{aligned} V &= \frac{1}{2}z^T Mz + \frac{1}{2}tr\tilde{W}_M^T Q_M \tilde{W}_M \\ &\quad + \frac{1}{2}tr\left(\tilde{W}_C^T Q_C \tilde{W}_C + \tilde{W}_H^T Q_H \tilde{W}_H\right) \end{aligned} \quad (16)$$

where Q_M , Q_C and Q_H are positive fixed weight matrix. Then, the derivative of V, which shows

$$\begin{aligned} \dot{V} &= z^T M\dot{z} + \frac{1}{2}z^T \dot{M}z + tr\tilde{W}_M^T Q_M \dot{\tilde{W}}_M \\ &\quad + tr\left(\tilde{W}_C^T Q_C \dot{\tilde{W}}_C + \tilde{W}_H^T Q_H \dot{\tilde{W}}_H\right) \end{aligned} \quad (17)$$

Utilizing the property: $\dot{M}(q) - 2C(q, \dot{q})$ are the skew symmetric matrix, and equation (17) becomes:

$$\begin{aligned} \dot{V} &= z^T M\dot{z} + z^T Cz + tr\tilde{W}_M^T Q_M \dot{\tilde{W}}_M \\ &\quad + tr\left(\tilde{W}_C^T Q_C \dot{\tilde{W}}_C + \tilde{W}_H^T Q_H \dot{\tilde{W}}_H\right) \end{aligned} \quad (18)$$

The ideal weight matrix W_M , W_C and W_H are constant matrices, the following relationship can be implemented:

$$\begin{aligned} \dot{\tilde{W}}_M &= \dot{\hat{W}}_M \\ \dot{\tilde{W}}_C &= \dot{\hat{W}}_C \\ \dot{\tilde{W}}_H &= \dot{\hat{W}}_H, \end{aligned} \quad (19)$$

substituting (15) into the equation (18), then we have:

$$\begin{aligned} \dot{V} &= -z^T Kz \\ &\quad - tr\left[\tilde{W}_M^T \left(S_M(q)\dot{q}_r z^T + Q_M \dot{\tilde{W}}_M\right)\right] \\ &\quad + tr\left[\tilde{W}_C^T \left(S_C(q, \dot{q})q_r z^T + Q_C \dot{\tilde{W}}_C\right)\right] \\ &\quad + tr\left[\tilde{W}_H^T \left(S_H(q)z^T + Q_H \dot{\tilde{W}}_H\right)\right] \end{aligned} \quad (20)$$

According to [25], the upgraded principle is given as follow,

$$\begin{aligned} \dot{\hat{W}}_M &= -Q_M^{-1} \left(S_M(q)\dot{q}_r z^T + \sigma_M \hat{W}_M\right) \\ \dot{\hat{W}}_C &= -Q_C^{-1} \left(S_C(q, \dot{q})q_r z^T + \sigma_C \hat{W}_C\right) \\ \dot{\hat{W}}_H &= -Q_H^{-1} \left(S_H(q)z^T + \sigma_H \hat{W}_H\right) \end{aligned} \quad (21)$$

where σ_M , σ_C , σ_H are pre-designed positive constants.

Combining (21) and (20), then:

$$\begin{aligned} \dot{V} &= -z^T Kz - \sigma_M tr\left(\tilde{W}_M^T \hat{W}_M\right) \\ &\quad - \sigma_C tr\left(\tilde{W}_C^T \hat{W}_C\right) - \sigma_H tr\left(\tilde{W}_H^T \hat{W}_H\right) \end{aligned} \quad (22)$$

Applying Young's inequality, (22) can be extended to

$$\begin{aligned} \dot{V} &= -z^T Kz + \frac{\sigma_M tr(W_M^{*T} W_M^*)}{2} - \frac{\sigma_M tr(\tilde{W}_M^T \tilde{W}_M)}{2} \\ &\quad + \frac{\sigma_C tr(W_C^{*T} W_C^*)}{2} - \frac{\sigma_C tr(\tilde{W}_C^T \tilde{W}_C)}{2} \\ &\quad + \frac{\sigma_H tr(W_H^{*T} W_H^*)}{2} - \frac{\sigma_H tr(\tilde{W}_H^T \tilde{W}_H)}{2} \end{aligned} \quad (23)$$

Finally:

$$\dot{V} \leq 0 \quad (24)$$

As indicated by Lyapunov stability theorem, the closed-loop stability is guaranteed.

VI. EXPERIMENTAL STUDIES

A. EXPERIMENT SETUP

In these experimental studies, series of tests were set up to examine the accuracy of the designed control system as shown in Fig.5, using NN learning. The Leap Motion API & SDKs based detector tracked and provided the coordinate data of all the five fingertips, the inverse kinematics calculated all the corresponding three joint angle values of each finger. UDP was employed to send all the 15 set of finger joint values q_d to the NN based adaptive controller written in the Simulink function of MATLAB to control the robot hand. In the first experiment, the operator smoothly moved his fingers by putting his hand over the LeapMotion (shown in Fig.2). At the same time, the angular values of the finger joints were outputted to a local file for recording purposes. The motion space of the DLR-HIT II robot hand is different from the one of LeapMotion. Therefore, we need to do a linear space transformation to project the spatial coordinates of LeapMotion into the spatial coordinates of the robot hand. The linear relationship can be expressed as:

$$\theta_{DLR} = T\theta_{LeapMotion} + B \quad (25)$$

where T is the transformation matrix and B is the deviation matrix. The linearly transformed reference trajectory q_d served as the reference trajectory of the robot hand and the input signal of the controller. In this experiment, the NN based adaptive controller was not applied.

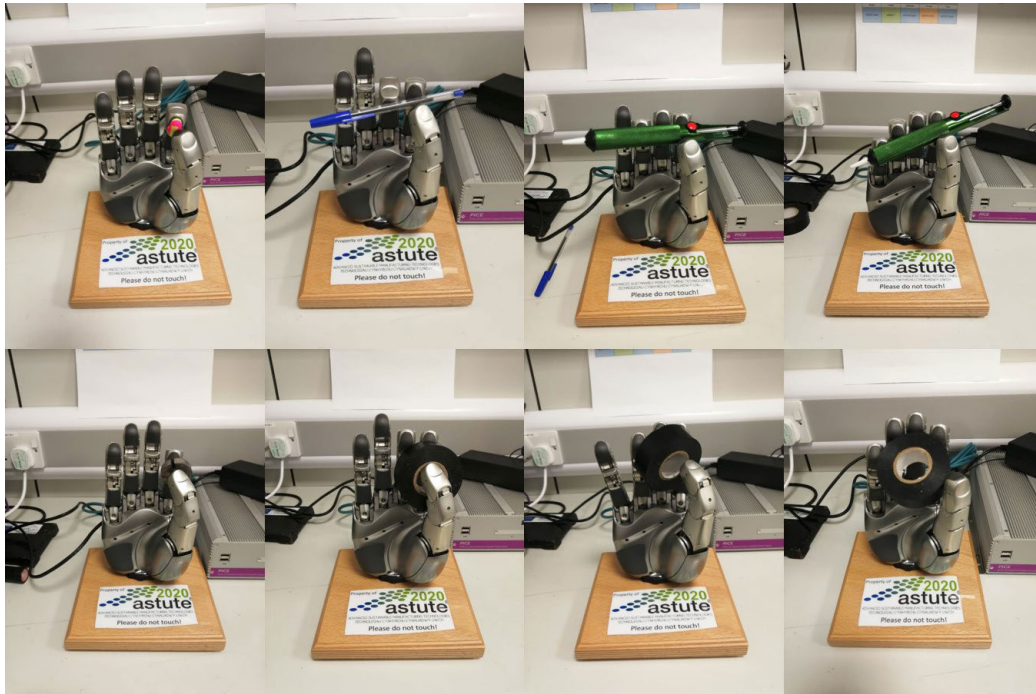


FIGURE 5. Graphical demonstration of the grasping tasks using 2,3,4,5 fingers.

For the second experiment, a heavy object was organized to be put on the area that the DLR Hand II can reach and grasp using 2, 3, 4, 5 fingers respectively. Meanwhile, the operator, stood in front of the robot hand and remotely controlled the robot hand for fingers closing/opening tasks by putting his hand over the Leap Motion sensor and performing grasping/opening behaviours of his hand with 2, 3, 4, 5 fingers respectively. In addition, in this experimental work, we defined the stiffness of each robot hand finger as $500(N/m)$ because in this level, the objects been grasped was not easily fallen down or neither damaged. The reference trajectory q_d was linearly converted and used as the input signal of the controller. The controller based on the RBFNNs function approximation made the actual trajectory of the DLR-HIT II robot hand keep up with the trajectory of the hand of the human operator. In this experiment, the operator completed a periodic set of grasping actions at a steady speed. We mainly tested the tracking effects of the finger's joints without the ones in the palm: PIP joints, DIP joints, MCP joints, CMC joints and IP joints. First, the LeapMotion camera space was transformed into the motion space of the DLR-HIT II robot hand by the linear transformation relationship of equation (25), where more detailed methodology of the coordination system transformation can be found in our previous work [28]. Then, the remote master computer sent the transformed track signal q_d to the slave computer through UDP to control the robot hand. In the torque mode of the DLR-HIT II robot hand, the reference trajectories signals q_d were used as the input of the controller, which were also the joints angles of real-time transformations from LeapMotion.

Moreover, the joints angles of the robot hand were used as the feedback signals.

In order to better understand the sensitivity of our proposed approach to the parameters and constants, the third experiment was conducted to investigate the sensitivity of the performance relative to all parameters, which are the control gain K ; RBFNN parameters Q and σ ; the initial values of the weight matrix: \hat{W}_M^T , \hat{W}_C^T , \hat{W}_H^T ; the number of nodes in the neural network l ; the variance of Gaussian function ζ , respectively. Firstly, the averaged joint angular difference among all the 4 joints on each finger between the robot hand and the human hand was selected as the performance index. Then the changes of the performance index along the parameters tuning was studied to find the corresponding parameter values when the performance index has the lowest value. The recorded data from the first experiment were directly imported into the controller system via MATLAB. We made each parameter changes with the same ratio for 5 times in the vicinity of the nominal value in turn, and kept the other parameters remain unchanged, to observe the change of the performance index curve. Finally, the particular parameter where the control performance is more sensitive to, can be defined. Furthermore, the values of all corresponding parameters can be obtained in the case of the highest curve fit.

B. EXPERIMENTAL RESULTS

For the first experimental result analysis, Fig. 6 illustrates the experimental data of the joint angular variables for the DLR-HIT II robot hand while performing the smooth

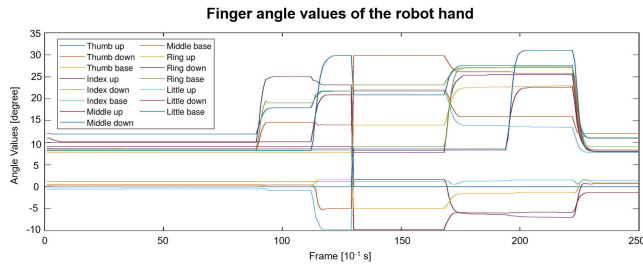


FIGURE 6. Joint values for all the five fingers of the robot hand while grasping using all the 5 fingers.

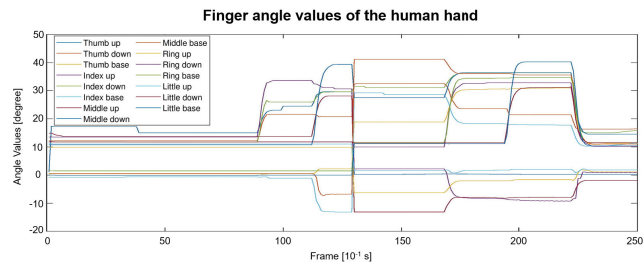


FIGURE 7. Finger joints values for the hand of the human operator collected from LeapMotion.

movement using all the 5 fingers, which are the robot hand’s real trajectory after the calculation of the inverse kinematics. It is easy to find that when the operator maintains a static posture, the trend of each joint angle remains relatively stable. The main reason for some jitter in the curve is that it is difficult for the operator to ensure that the hand is completely static and the data acquisition of the LeapMotion is not completely stable during the data acquisition process. As the comparison, Fig. 7 illustrates the angular trajectory collected from LeapMotion, which represents the finger’s joints angular variables for the hand of the human operator. From above two plotted graphs, it can be seen that even there are tracking errors existing, our proposed LeapMotion sensor guided teleoperation system without NN learning can track the human operator’s hand motion, make the DLR-HIT II robot hand imitate the human operator’s hand motion in a fair level.

There are dynamic uncertainties existing in the teleoperation process. To overcome this issue, RBFNNs were employed to approximate the unknown dynamics in this paper. During the grasping tasks, it can be found that even the stiffness parameters have been set correctly, using 2,3,4 fingers to stably grasp the objects were not easy to achieve. After plenty of trainings, we noticed that using all the 5 fingers had the best grasping stabilities. Therefore, all 5 fingers were used to conduct grasping tasks for the second experiment. In addition, because the proximal interphalangeal (PIP) joint among all the joints of each finger was most bent while grasping items, here only the PIP joints of all the 5 fingers were taken for data analysis to the adaptive control process in this experiment, for the easy-to-see experimental data curves and analysis as well, which were representing the *ThumbUp*, *IndexUp*, *MiddleUp*, *RingUp* and *LittleUp* joints, respectively. For our second experiment, even though all the data have been collected when performing the grasp tasks using 2,3,4,5 fingers, respectively, it is still needed to plot the data set for just the 5 fingers’ motion in order to taking the visual duplicate problems into account. Fig. 8 shows the comparison of all the 5 fingers’ PIP joints controlled with NN learning and without NN learning. It can be seen that the tracking performance of the fingers’ joints have been improved after adopting the NN learning control. As can be seen from Fig. 8, there is external torque applied to all 5 fingers when the DLR-HIT II robot hand grasps the heavy objects. As a consequence, the angular trajectory of the robot hand differs from the one of human operator when the only the LeapMotion controller is employed. While the proposed RBFNNs based adaptive controller is employed together with the LeapMotion tele-control system, in order to adapt to the external torque, the robot hand generates a corrected reference trajectory q_r by correcting the previous reference trajectory q_d . In addition, the actual curve of the joint can converge to the reference trajectory curve better. The main reason for the error between the curves is the delay of UDP data transmission and the jitter during the movement of the robot hand. Finally, it can be concluded that the controller has made the robot hand accurately track the corrected reference track q_r .

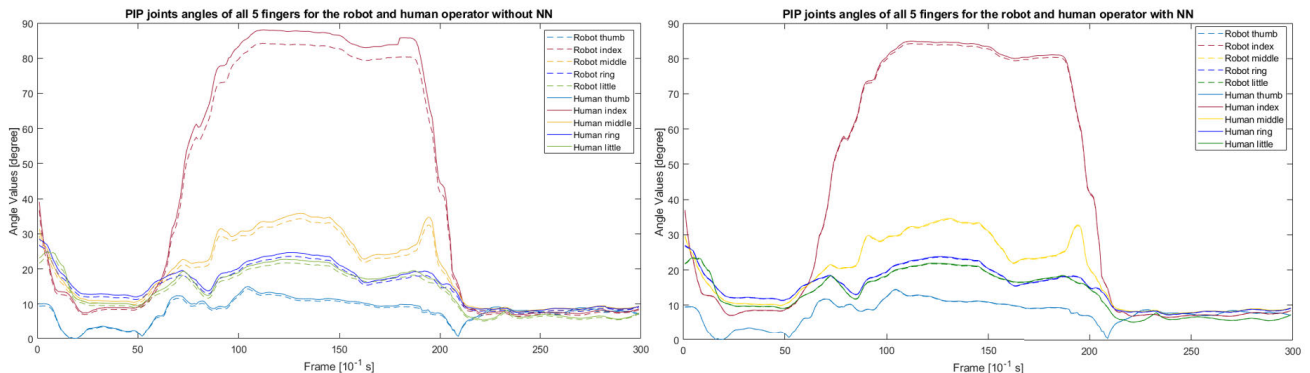


FIGURE 8. PIP joints variables among all the 5 fingers between the human operator and the robot for both with NN and without NN.

TABLE 3. Performance index table under different variables of the control gain K and the RBFNN parameter Q.

Parameter type	Control Gain K					RBFNN parameter Q				
Parameter value	10	11	12	13	14	830	915	1000	1085	1170
PI thumb [degree]	0.11165	0.10459	0.09874	0.10798	0.11985	0.20532	0.15532	0.09874	0.18532	0.22654
PI index [degree]	0.10865	0.09789	0.08987	0.10685	0.11886	0.19887	0.14565	0.08987	0.18865	0.2231
PI middle [degree]	0.10369	0.09212	0.07298	0.10523	0.10393	0.16565	0.12216	0.07298	0.1598798	0.21965
PI ring [degree]	0.10493	0.09423	0.08784	0.10647	0.11739	0.19854	0.14446	0.08784	0.17646	0.21625
PI little [degree]	0.10406	0.09256	0.07897	0.10596	0.10759	0.17984	0.13526	0.07897	0.16979	0.21271

TABLE 4. Performance index table under different variables of the RBFNN parameter σ and the Gaussian variance ζ .

Parameter type	RBFNN parameter sigma					Gaussian variance varsigma				
Parameter value	0.008	0.009	0.01	0.011	0.012	80	90	100	110	120
PI thumb [degree]	0.17425	0.12817	0.09874	0.16304	0.20929	0.13271	0.10296	0.09874	0.10517	0.1116
PI index [degree]	0.16912	0.12404	0.08987	0.15871	0.20584	0.13598	0.10252	0.08987	0.10473	0.11959
PI middle [degree]	0.16399	0.11991	0.07298	0.15439	0.20239	0.13926	0.10207	0.07298	0.104288	0.135596
PI ring [degree]	0.15887	0.11578	0.08784	0.15006	0.19894	0.14253	0.10163	0.08784	0.10384	0.11985
PI little [degree]	0.15374	0.11165	0.07897	0.14574	0.19549	0.1458	0.10119	0.0789	0.1034	0.12783

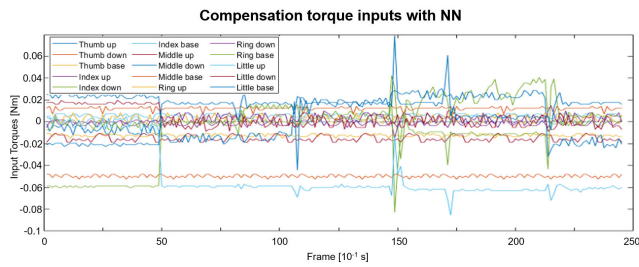


FIGURE 9. Compensation torque inputs of the control system with NN.

According to our large numbers of parameters tuning based on the experimental study, the control system had the greatest performance when the parameters are set as: control gain $K = 12$; RBFNN parameters $Q = 10^3$, $\sigma = 10^{-2}$; the initial value of the weight matrix: $\hat{W}_M^T = \hat{W}_C^T = \hat{W}_H^T = 0 \in R^{n \times l}$; the number of nodes in the neural network $l = 3$; the variance of Gaussian function $\zeta = 100$, wherein the initial value of the weight matrix and the NN nodes numbers are set according to the experiences based knowledge from our previous research works.

From Tab.3, we can observe that the performance index (PI), which is the averaged joint angular difference among all the 4 joints on each finger between the robot hand and the human hand, changes slightly when the parameter K is perturbed from 10 to 14; the performance index changes fairly when the RBFNN parameter Q is perturbed from 830 to 1170. From Tab.4, we can observe that the performance index changes fairly when the RBFNN parameter σ is perturbed from 0.008 to 0.012; the performance index changes slightly when the parameter Gaussian variance ζ is perturbed from 80 to 120. Overall, the control performance is more sensitive to the RBFNN parameters Q and σ . How to improve the robustness of the proposed control system against these parameters, and reduce the overall sensitivity of the method relative to parameter perturbations will be systematically investigated in our future work.

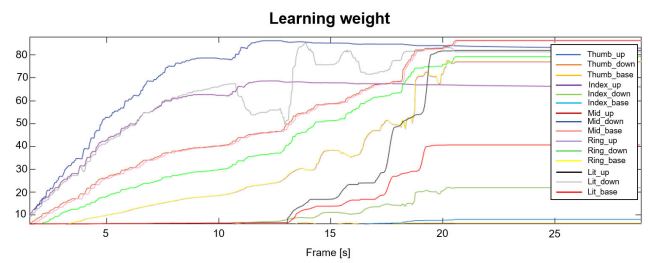


FIGURE 10. NN learning weights of every single joint.

Fig.9 shows the joint compensation torque curves. The output torque of the NN is mainly to compensate for the uncertainties in the system to achieve the desired control performance. It shows that the uncertainty of the control system appears at certain moments after the application of the neural network, and the control system can be well compensated by the input torque. The NN learning weights can be seen in Fig.10. As the stiffness has been set to a constant value, the angular value difference of finger joints between the operator and the robot hand are selected to assess the control performance. It can be concluded from the above results that NN learning promotes the robustness of the tele-control system as it has an acceptable error margin during the teaching and learning process, thereby satisfying the proposed design goal.

VII. CONCLUSION

In this work, an NN-based adaptive teleoperation system for the DLR-HIT II robot hand has been designed using the Leap Motion hand tracking device. Firstly, the Leap Motion device outputs the 3D Cartesian coordinates of each joint of the operator's hand, then inverse kinematics are used to obtain the relevant joint angles for the robot hand. Next, these joint angles are sent via the UDP communication protocol to the controller which subsequently outputs joint angles to the robotic hand.

In order for the robot to completely do the motion along with the desired trajectory q_d and to overcome the existence of dynamic uncertainties, the experimental work was conducted to compare the effect of the unknown payload under NN learning with that without NN learning. The results illustrate that the NN-learning applied teleoperation system achieves the control of the DLR-HIT II robot hand with the smallest number of errors.

REFERENCES

- [1] S. Ma, "Integration technology for advanced manufacturing: R&D in China 863 program," in *Proc. IEEE Int. Conf. Integr. Technol.*, Mar. 2007, p. null.
- [2] S. A. Green, M. Billinghurst, X. Chen, and J. G. Chase, "Human-robot collaboration: A literature review and augmented reality approach in design," *Int. J. Adv. Robot. Syst.*, vol. 5, no. 1, p. 1, Mar. 2008.
- [3] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: A survey," *Found. Trends Hum.-Comput. Interact.*, vol. 1, no. 3, pp. 203–275, 2008.
- [4] C. Bartneck and J. Forlizzi, "Shaping human-robot interaction: Understanding the social aspects of intelligent robotic products," in *CHI Extended Abstracts on Human Factors in Computing Systems*. New York, NY, USA: ACM, 2004, pp. 1731–1732.
- [5] R. H. Taylor, "A perspective on medical robotics," *Proc. IEEE*, vol. 94, no. 9, pp. 1652–1664, Sep. 2006.
- [6] A. M. Okamura, "Methods for haptic feedback in teleoperated robot-assisted surgery," *Ind. Robot, Int. J.*, vol. 31, no. 6, pp. 499–508, 2004.
- [7] Y. Nakabo and M. Ishikawa, "Visual impedance using 1 ms visual feedback system," in *Proc. IEEE Int. Conf. Robot. Automat.*, vol. 3, May 1998, pp. 2333–2338.
- [8] R. Volpe and P. Khosla, "A theoretical and experimental investigation of explicit force control strategies for manipulators," *IEEE Trans. Autom. Control*, vol. 38, no. 11, pp. 1634–1650, Nov. 1993.
- [9] L. Huang, S. S. Ge, and T. H. Lee, "Fuzzy unidirectional force control of constrained robotic manipulators," *Fuzzy Sets Syst.*, vol. 134, no. 1, pp. 135–146, 2003.
- [10] H. Mehdi and O. Boubacker, "Stiffness and impedance control using Lyapunov theory for robot-aided rehabilitation," *Int. J. Social Robot.*, vol. 4, no. 1, pp. 107–119, 2012.
- [11] S. Seshagiri and H. K. Khalil, "Output feedback control of nonlinear systems using RBF neural networks," *IEEE Trans. Neural Netw.*, vol. 11, no. 1, pp. 69–79, Jan. 2000.
- [12] C. Cecati, J. Kolbusz, P. Rózycki, P. Siano, and B. M. Wilamowski, "A novel RBF training algorithm for short-term electric load forecasting and comparative studies," *IEEE Trans. Ind. Electron.*, vol. 62, no. 10, pp. 6519–6529, Oct. 2015.
- [13] M. Robnik-Šikonja, "Data generators for learning systems based on RBF networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 5, pp. 926–938, May 2016.
- [14] R. Yang, P. Er, Z. Wang, and K. Tan, "An RBF neural network approach towards precision motion system with selective sensor fusion," *Neurocomputing*, vol. 199, pp. 31–39, Jul. 2016.
- [15] T. Mouri, H. Kawasaki, K. Yoshikawa, J. Takai, and S. Ito, "Anthropomorphic robot hand: Gifu hand III," in *Proc. Int. Conf. ICCAS*, 2002, pp. 1288–1293.
- [16] J. Butterfaß, M. Grebenstein, H. Liu, and G. Hirzinger, "DLR-Hand II: Next generation of a dextrous robot hand," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, vol. 1, May 2001, pp. 109–114.
- [17] H. Liu, P. Meusel, N. Seitz, B. Willberg, G. Hirzinger, M. H. Jin, Y. W. Liu, R. Wei, and Z. W. Xie, "The modular multisensory DLR-HIT-hand," *Mech. Mach. Theory*, vol. 42, no. 5, pp. 612–625, May 2007.
- [18] Z. Chen, N. Y. Lii, T. Wimböck, S. Fan, H. Liu, and A. Albu-Schäffer, "Experimental analysis on spatial and Cartesian impedance control for the dexterous DLR/HIT II hand," *Int. J. Robot. Automat.*, vol. 29, no. 1, pp. 1–13, 2014.
- [19] G. Zhang, Y. Sun, J. Wang, Z. Gan, H. Zhang, and G. Rossano, "Accelerometer to monitor movement of a tool assembly attached to a robot end effector," U.S. Patent 11 051 383, Feb. 4, 2005.
- [20] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and Kinect devices," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 1565–1569.
- [21] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1188–1192, Sep. 2016.
- [22] L. E. Potter, J. Araullo, and L. Carter, "The leap motion controller: A view on sign language," in *Proc. 25th Austral. Comput.-Hum. Interact. Conf., Augmentation, Appl., Innov., Collaboration*, 2013, pp. 175–178.
- [23] A. Lamikiz, L. L. De Lacalle, O. Ocerin, D. Díez, and E. Maidagan, "The denavit and hartenberg approach applied to evaluate the consequences in the tool tip position of geometrical errors in five-axis milling centres," *Int. J. Adv. Manuf. Technol.*, vol. 37, nos. 1–2, pp. 122–139, 2008.
- [24] C. Yang, G. Peng, L. Cheng, J. Na, and Z. Li, "Force sensorless admittance control for teleoperation of uncertain robot manipulator using neural networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published.
- [25] C. Yang, J. Chen, and F. Chen, "Neural learning enhanced teleoperation control of baxter robot using IMU based motion capture," in *Proc. 22nd Int. Conf. Automat. Comput. (ICAC)*, 2016, pp. 389–394.
- [26] Y. Zhao and C. C. Cheah, "Neural network control of multifingered robot hands using visual feedback," *IEEE Trans. Neural Netw.*, vol. 20, no. 5, pp. 758–767, May 2009.
- [27] Y. Fang, K. A. Loparo, and X. Feng, "Inequalities for the trace of matrix product," *IEEE Trans. Autom. Control*, vol. 39, no. 12, pp. 2489–2490, Dec. 1994.
- [28] C. Li, C. Yang, J. Wan, A. S. K. Annamalai, and A. Cangelosi, "Teleoperation control of baxter robot using Kalman filter-based sensor fusion," *Syst. Sci. Control Eng.*, vol. 5, no. 1, pp. 156–167, 2017.



CHUNXU LI received the B.Eng. degree in chemical engineering from the Qingdao University of Science and Technology, Qingdao, China, in 2014, and the M.Sc. degree in chemical engineering and the Ph.D. degree in electrical and electronics engineering from Swansea University, Swansea, U.K., in 2016 and 2019, respectively. He is currently a Postdoctoral Project Officer of Advanced Sustainable Manufacturing Technologies 2020 with Swansea University and also a Visiting Associate Professor with the School of Automation and Electrical Engineering, Qingdao University of Science and Technology. His research interests include robotics, automation, and computational intelligence.



ASHRAF FAHMY has expertise in decision making, manufacturing systems, robotic manipulation, instrumentation, control systems, and electrical power generation. He is currently the Senior Technical Manager of ASTUTE with the Engineering College, Swansea University, an Associate Professor with Helwan University, and the HV Manager of the Shaker Consultancy Group. He is an electrical power and machines drives engineer by education and worldwide experience, robotics control engineer by research, and industrial manufacturing consultant by U.K. and worldwide experience.



JOHANN SIENZ received the B.Eng. (Hons.) from the University of Central Lancashire, U.K., in 1989, the Diplom-Ingenieur (FH) degree in mechanical engineering in aerospace engineering from the University of Applied Sciences Augsburg, Germany, in 1989, and the M.Sc. and Ph.D. degrees from Swansea University, U.K., in 1990 and 1994, respectively. He is the Personal Chair of the College of Engineering, Swansea University, where he is the Deputy Head of College, and the Director of Innovation and Engagement. He is a chartered engineer (CEng) and a chartered mathematician (CMath). He is a Fellow of the Institution of Mechanical Engineers (FIMechE) and the Royal Aeronautical Society (FRAeS). He is a member of the Institute of Mathematics and its Applications (MIMA). He is the Editor-in-Chief of the international journal *Applied Mathematical Modelling*. He is on the editorial board of several other international journals.

...