Received: 29 January 2018 | Revised: 6 May 2018 | Accepted: 14 May 2018

DOI: 10.4218/etrij.2018-0057

SPECIAL ISSUE

WILEY ETRI Journal

# Work chain-based inverse kinematics of robot to imitate human motion with Kinect

**Ming Zhang** | **Jianxin Chen** | **Xin Wei** | **Dezhou Zhang**

Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing, China.

**Correspondence**
Jianxin Chen, Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education, Nanjing, China.
Email: chenjx@njupt.edu.cn

**Funding information**
Key Lab of Broadband Wireless Communication and Sensor Network Technology; Nanjing University of Posts and Telecommunications, Ministry of Education, Grant/Award Number: JZNY201704; Nanjing University of Posts and Telecommunications, Grant/Award Number: NY217021; National Natural Science Foundation of China, Grant/Award Number: 61401228; Natural Science Foundation of Jiangsu Province, Grant/Award Number: BK20140891.

The ability to realize human-motion imitation using robots is closely related to developments in the field of artificial intelligence. However, it is not easy to imitate human motions entirely owing to the physical differences between the human body and robots. In this paper, we propose a work chain-based inverse kinematics to enable a robot to imitate the human motion of upper limbs in real time. Two work chains are built on each arm to ensure that there is motion similarity, such as the end effector trajectory and the joint-angle configuration. In addition, a two-phase filter is used to remove the interference and noise, together with a self-collision avoidance scheme to maintain the stability of the robot during the imitation. Experimental results verify the effectiveness of our solution on the humanoid robot Nao-H25 in terms of accuracy and real-time performance.

**KEYWORDS**
humanoid robot, imitation, inverse kinematics, motion tracking, work-chains

## 1 | INTRODUCTION

In recent times, humanoid robots have attracted much attention owing to the wide application of the robot community on applications such as telemedicine, home service, and distance education. Among them, human-motion imitation is currently a popular topic, and is considered a good candidate for studying the principle of mechanical control.

There have been several studies on the imitation of robot motion imitation. Originally, human motion was captured and optimized off-line to adapt to the robot structure and constraints [1–3]. Recently, real-time imitation systems [4–13] have been developed owing to the development of three-dimensional (3D) motion-tracking equipment such as Kinect [14] and Xsens MVN [15]. However, owing to the differences in the joints of the human body and robot, there is a discussion about the motion similarity during mapping motions from humans to robots [16,17]. Generally, there are two definitions for motion similarity: one is the similarity between end-effector trajectories [8–13], and the other is the similarity between angular configurations [4–7]. Both definitions have been used for robot control.

Although these studies have solved some problems related to humanoid robot imitation, there remain several

challenges. For example, the imitation should be implemented in real time. Generally, the human skeletal data obtained from a Kinect sensor should be smoothed to minimize jitter due to the noise and interference. This processing should be completed within a short period. After that, the processed motion data can be sent to the robot. Meanwhile, owing to the differences in the joints of the human body and robot, it is not easy for robots to imitate humans with high accuracy. Moreover, during the imitation procedure, it is very important to maintain the stability of the robot. To address this issue, in this paper, we propose a system that enables a robot to imitate upper-body motions in real time. The main contributions of this paper are as follows:

1) To simplify the inverse kinematics process, we built two work chains on each arm. This method not only guarantees the similarity of the end-effector trajectories, but also increases the similarity of angle configurations.
2) We used a two-phase filter to remove the interference and noise contained in the joint angles to make the imitation more stable.
3) Many experiments have been performed to verify the effectiveness of our solution in terms of motion similarity and imitation delay.

The rest of this paper is organized as follows. Section 2 introduces the related work on the robot motion imitation, and Section 3 describes the framework of our imitation system. Section 4 introduces the implementation of the system in detail. Section 5 analyzes the performance and Section 6 concludes the paper.

## 2 | RELATED WORK

To map motions from a human to robot, there are two types of motion similarity: the similarity between the end-effector trajectories, and the similarity between the angular configurations. For the latter, Ningjia and others [4] proposed a geometric method to calculate the upper body joint angles of robots using human skeleton data captured with a Kinect sensor [18]. Lemtwally and others [5] used the spatial vector method to obtain all joint angles of robot. Zuher and others [6] proposed three mathematical methods for the joint-angle calculation. Chen and others [7] proposed a lower-body imitation strategy that is based on the geometry to guarantee the stability of the robot. All of them used the Kinect sensor [14] to capture human motion.

However, Kruger and others [16] and Tang and others [17] claimed that the similarity between the end-effector trajectories is more effective than that between the angular configurations. Since then, there have been some studies on the end-effector similarity. Among them, the inverse kinematic (IK) algorithm is an important algorithm to achieve the similarity between the end-effector trajectories. A classic IK algorithm is the weighted least-norm (WLN) algorithm [19], which uses a weight matrix to avoid joint-angle limitations. The other one is the damped least-squares (DLS) algorithm [18], which is similar to the Levenberg-Marquardt (LM) algorithm [20]. Besides that, Nikos and others [21] presented a forward-kinematics equation and an IK analytical solution for the Nao robot.

In addition, there have been some IK-based algorithms for the robot imitation. Koenemann and others [8,9] used a DLS-based IK solver to solve the IK problem for the real-time whole-body motion imitation by the Nao robot. They used the special equipment-MVN suit [15] to track human motion because it provides more accurate motion information than Kinect. Fan Wang and others [10] proposed a new IK algorithm to achieve imitation. They transformed the IK solution into an optimization problem that was solved using the LM algorithm. Liang Zhang and others [11] proposed a real-time whole-body imitation with the Nao robot. They combined the DLS algorithm and WLN algorithm to solve the IK problem. Shohin Mukherjee and others [12] used the iteration method and adaptive neuro-fuzzy inference systems to solve the IK. Alibiing and others [13] combined the angle-configuration similarity and the end-trajectory similarity. They calculated the joint angles using geometry before solving the IK problem.

Although there are several studies that focus on developing robots that imitate human motion, there remain some challenges, such as the ability to achieve real-time performance, imitation accuracy, and imitation stability. In the next section, we will present a new solution to address these issues.

## 3 | SYSTEM OVERVIEW

Figure 1 depicts the system model of a humanoid robot mimicking human motion in real time. Our implementation consists of six parts. First, the human skeleton data are captured by a Kinect sensor. The Kinect sensor provides 3D skeletal tracking information at 30 fps, and the frame data contains position information about 20 joints of the human body. However, owing to the occlusion, environmental noise, and interference, the human skeletal data should be smoothed to minimize jitter. Here, we used a smoothing filter in the Holt double exponential smoothing method to filter the skeletal data. Then, we used the vector information to describe the relative relationship between each joint instead of the location information. Meanwhile, the vectors will be mapped from the human space to the robot space owing to the joint difference.

After that, we set up two work chains on each hand, and the forward kinematics equation is built based on the

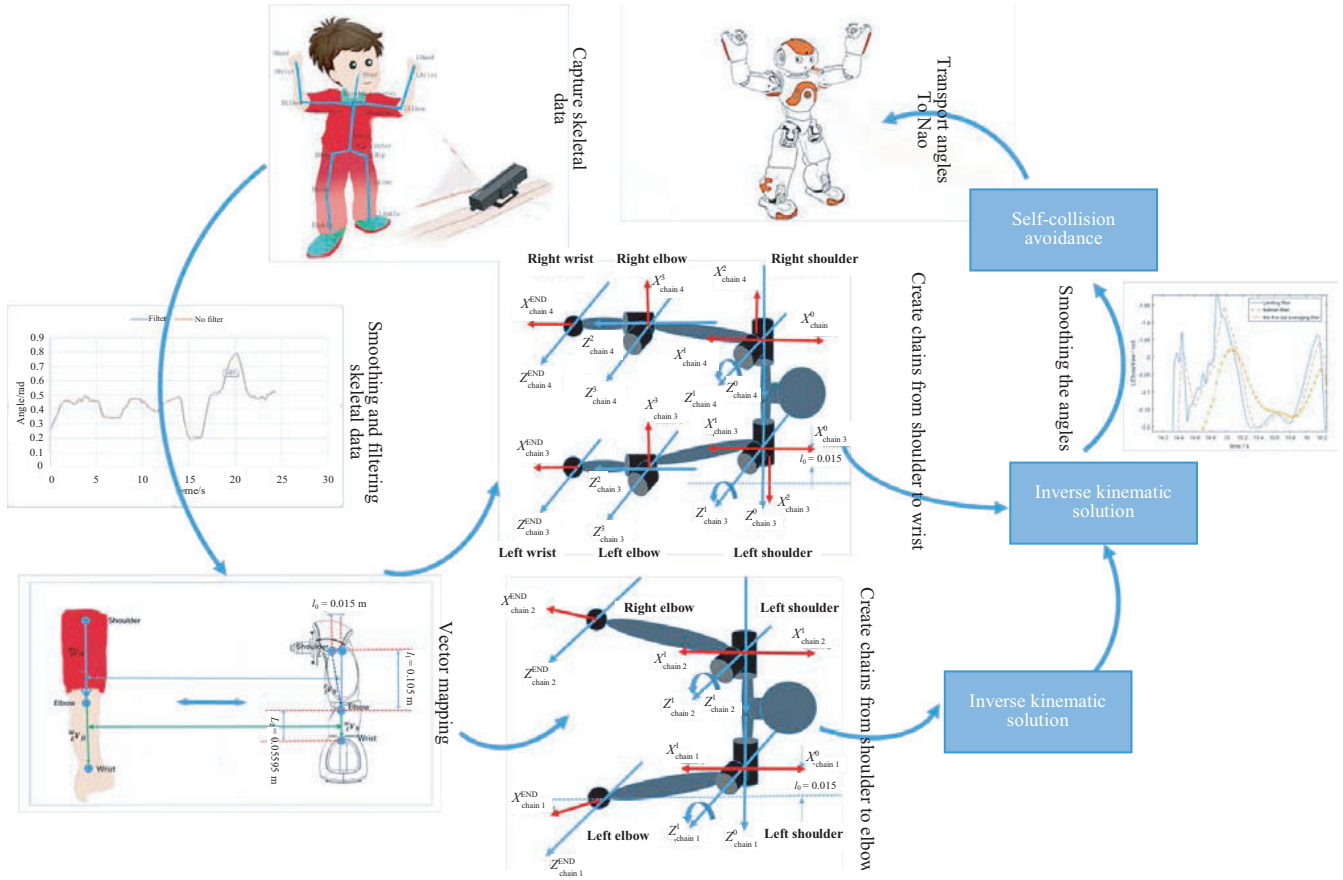


**FIGURE 1** Humanoid robot mimicking system

distal DH convention. These work chains not only simplify the IK solution process, but also ensure the similarity of end trajectories. Then, the robot joint angles are obtained by solving the IK of the work chains. To make the imitation more stable, we used a two-step filter to control the joint angles. Finally, we constrained the angle to prevent self-collisions of the humanoid robot. In Table 1, we list the notations that are used in this paper.

## 4 | WORK-CHAIN-BASED REVERSE KINEMATICS

In our implementation, we ensure real-time performance by simplifying the inverse kinematics solution to reduce the time complexity of the algorithm. Then, by building two work chains on each arm, we conquer the problem of joint difference to ensure the similarity of the joint trajectories. In addition, we used a two-step filter to realize control of the joint angles to maintain the stability of the robot during the imitation.

Figure 2 depicts four kinematic chains on the arms of the Nao robot. For each arm, there are two chains. The first kinematic chain extends from the shoulder to the elbow, and the second kinematic chain extends from the shoulder to the wrist.

### 4.1 | Vector mapping

To use the captured motion from the Kinect for the humanoid robot, we need to transform it into the space of the robot as the positions are available from the motion capture data. We used the vector information to describe the relative relationship between each joint instead of the location information.

For the left arm, let ${}^{\mathrm{E}}_{\mathrm{S}}V_{\mathrm{H}}$ denote the vector between the shoulder and the elbow of the human body, and let ${}^{\mathrm{E}}_{\mathrm{S}}V_{\mathrm{N}}$ denote the corresponding vector of the Nao robot. ${}^{\mathrm{E}}_{\mathrm{S}}V_{\mathrm{H}}$ denotes the vector between the shoulder and the wrist of the human body, and ${}^{\mathrm{W}}_{\mathrm{S}}V_{\mathrm{N}}$ represents the corresponding vector of the Nao robot, as in Figure 3. $l_0$, $l_1$, and $l_2$ represent the offsets of each joint or the size of the connecting rod, which were defined in the datasheet [22]. Then, we can map these vectors on the human to the robot using (1) and (2). For the right arm, the mapping equations are similar.

$$ {}^{\mathrm{E}}_{\mathrm{S}}V_{\mathrm{N}} = \frac{{}^{\mathrm{E}}_{\mathrm{S}}V_{\mathrm{H}}}{\left\| {}^{\mathrm{E}}_{\mathrm{S}}V_{\mathrm{H}} \right\|} * l_1 + \begin{bmatrix} 0 \\ l_0 \\ 0 \end{bmatrix}, \tag{1} $$

**TABLE 1** Definition of notations

| | |
|---|---|
| $S$ | Shoulder joint |
| $E$ | Elbow joint |
| $W$ | Wrist joint |
| ${}^{j}_{S}V_H$ | Vector between the shoulder and the joint $j$ in the human |
| ${}^{j}_{S}V_N$ | Mapping vector of ${}^{j}_{S}V_H$ in the robot |
| $l_0$, $l_1$, $l_2$ | Offset of each joint in the Nao robot |
| $\theta_0$, $\theta_1$, $\theta_2$, $\theta_3$ | Joint angles of the arm defined in Nao robot, the first two relate to the joints on the shoulder, and the last two relate to the joints of elbow |
| $x$ | ${}^{-1}\, l_0/l_1$ |
| ${}^{j}_{i-1}T$ | Transformation matrix from the joint $j$ to joint $j-1$ |
| Rot($z$,$\theta_i$) | Rotation matrix which rotates around the $z$ axis with $\theta_i$ degree |
| Rot($x$,$A_i$) | Rotation matrix which rotates around the $x$ axis with the degree of $A_i$ |
| Trans(0,0,$d_i$) | Translation matrix which translates along the $z$ axis with $d_i$ |
| Trans($a_i$,0,0) | Translation matrix which translates along the $x$ axis with $a_i$. |
| ${}^{K}_{S}R$ | Rotation matrix from the Kinect coordinate system to the shoulder coordinate system on the robot |
| ${}^{j}_{S}V$ | Vector from the shoulder to the joint $j$ in the coordinate system of Nao robot |
| $V$ | $[0\ 0\ 0\ 1]^T$ |
| ${}^{j}_{S}V^4$ | $[{}^{j}_{S}V^T 1]^T$ |
| ${}^{j}_{j-1}T_{ch\_i}$ | Transformation matrix from joint $j$ to joint $j-1$ in the work chain $i$ |

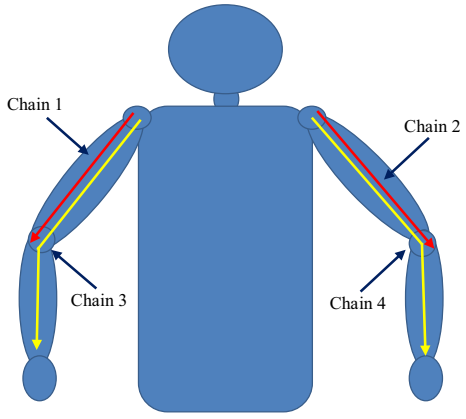


**FIGURE 2** Four positive kinematic chains on two hands

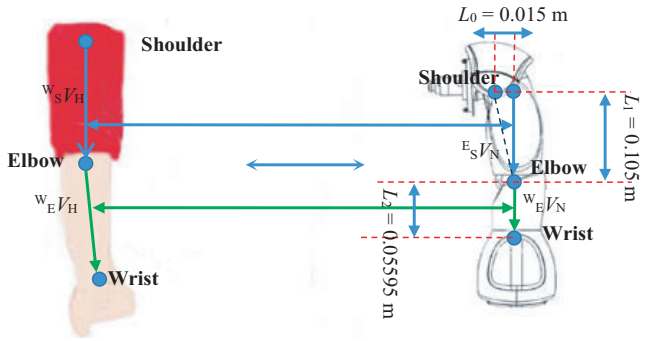


**FIGURE 3** Mapping vectors

$$ {}^{W}_{S}V_N = \frac{{}^{E}_{S}V_H}{\|{}^{E}_{S}V_H\|} * l_1 + \frac{{}^{W}_{E}V_H}{\|{}^{W}_{E}V_H\|} * l_2 + \begin{bmatrix} 0 \\ l_0 \\ 0 \end{bmatrix}. \quad (2) $$

## 4.2 | Forward kinematics using distal D-H parameters

The Denavit Hardenberg (D-H) parameters were used to assign a right-handed orthogonal coordinate system to each link in an open kinematic chain. The transformations between adjacent frames can be performed by a uniform coordinate transformation matrix. There are two main types of D-H conventions: distal and proximal [22]. Here, we used the distal D-H in our work chains. As opposed to previous studies in which one work chain is built on each arm [8] [13], we built two work chains on each arm to ensure the similarity of the joint trajectories. First, we established work chains from the shoulder to the elbow on each arm of the robot, as in Figure 4. Table 2 lists the D-H parameters for the first chain.

Here,

$$ x = \tan^{-1} \frac{l_0}{l_1}. $$

Then, we built other work chains from the shoulder to the wrist on the arms of the robot, as shown in Figure 5. Table 3 lists the DH parameters for the third chain. As the

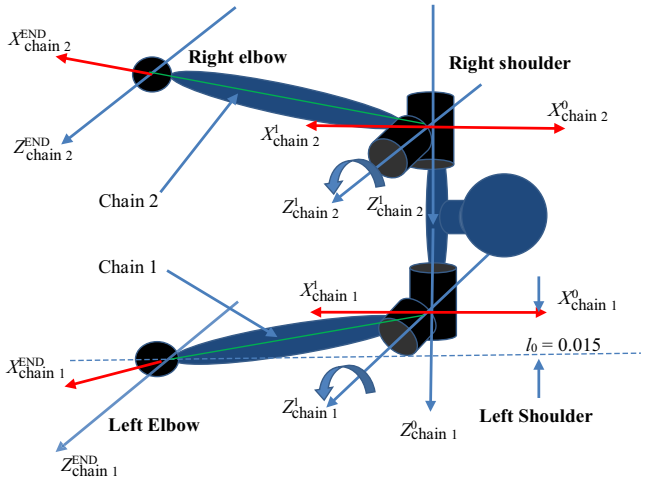


**FIGURE 4** Work chains from shoulder to elbow

**TABLE 2** DH parameters for work chain 1

| Chain1 | $\theta$ | $d$ | $a$ | $A$ |
|---|---|---|---|---|
| 0–1 | $\theta_0 + x$ | 0 | 0 | $\pi/2$ |
| 1–end | $\theta_1 + x$ | 0 | $\sqrt{{l_1}^2 + {l_0}^2}$ | 0 |

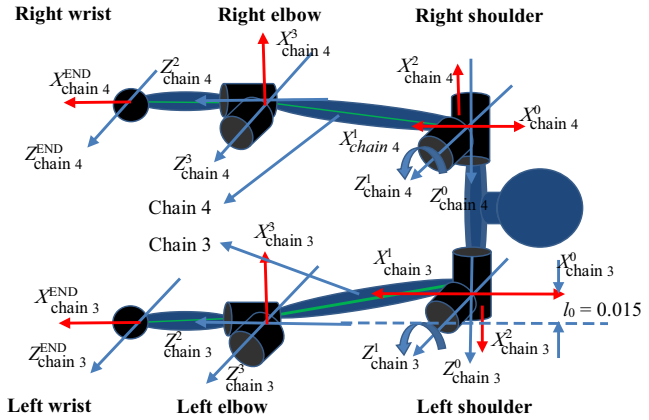


**FIGURE 5** Work chains from shoulder to wrist

**TABLE 3** DH parameters of chain 3

| Chain1 | $\theta$ | $d$ | $a$ | $A$ |
|---|---|---|---|---|
| 0–1 | $\theta_0 + \pi$ | 0 | 0 | $\pi/2$ |
| 1–end | $\theta_1 + \pi/2$ | 0 | $l_0$ | $\pi/2$ |
| 2–3 | $\theta_2 + \pi$ | $l_1$ | 0 | $\pi/2$ |
| 3–end | $\theta_3 + \pi/2$ | 0 | $l_2$ | 0 |

DH parameters for the two work chains of the right hand are similar to those of the left hand, we do not list them.

The DH transformation describes the translation and the orientation of joint $j$ based on the previous joint $j-1$ in the reference frame. The general DH matrix is

$$ {}^{j}_{j-1}T = \mathrm{Rot}(z,\theta_i)\cdot\mathrm{Trans}(0,0,d_i)\cdot\mathrm{Trans}(a_i,0,0)\cdot\mathrm{Rot}(x,A_i), \tag{3} $$

where $\mathrm{Rot}(z,\theta_i)$ is a rotation matrix that rotates around the $z$ axis at $\theta_i$ degree, and $\mathrm{Rot}(x,A_i)$ is a rotation matrix that rotates around the $x$ axis with an angle of $A_i$. $\mathrm{Trans}(0,0,d_i)$ is a translation matrix that translates along the $z$ axis with $d_i$, and $\mathrm{Trans}(a_i,0,0)$ is a translation matrix that translates along the $x$ axis with $a_i$. Then, the transformation matrix for chain $i$ can be denoted using the DH matrix as

$$ {}^{\mathrm{end}}_{0}T = {}^{1}_{0}T\cdot{}^{2}_{1}T\cdots{}^{\mathrm{end}}_{3}T. \tag{4} $$

## 4.3 | Inverse kinematic solution

As the robot is controlled using the configurations of the joint angles, we need to use the IK solution to compute the angles according to the vectors (1) and the transformation matrix (4). First, we need to map the vectors from the Kinect coordinate to the Nao coordinate. Equations (5) and (6) denote the rotation matrix and the position of the end effector in the base coordinate of the joint chain, respectively. Here, ROT(90°) denotes the rotation 90° around the $Y$ axis of the Kinect coordinate. ${}^{\mathrm{K}}_{\mathrm{S}}R$ is the rotation matrix of the shoulder from the Kinect coordinate to the robot coordinate. ${}^{j}_{\mathrm{S}}V$ is the vector from the shoulder to the joint $j$ in the shoulder coordinate of Nao. If $j$ is replaced with $W$, it denotes the wrist joint. If $j$ is replaced with $E$, it denotes the elbow joint.

$$ {}^{\mathrm{K}}_{\mathrm{S}}R = \mathrm{Rot}(Y, 90^\circ), \tag{5} $$

$$ {}^{j}_{\mathrm{S}}V = {}^{\mathrm{K}}_{\mathrm{S}}R\cdot{}^{j}_{\mathrm{S}}V_{\mathrm{Nao}}. \tag{6} $$

For chain 1 and chain 3 on the left arm, we can obtain the transformation matrix using Tables 2, 3, and (4). Then, the position of the end effector in the coordinates of the shoulder can be obtained using (6). In chain 1, the relationship between the position of the end joint and the joint angles can be represented as

$$ \begin{aligned} {}^{\mathrm{E}}_{\mathrm{S}}V^4 &= \begin{bmatrix} {}^{\mathrm{E}}_{\mathrm{S}}X & {}^{\mathrm{E}}_{\mathrm{S}}Y & {}^{\mathrm{E}}_{\mathrm{S}}Z & 1 \end{bmatrix}^T \\ &= {}^{1}_{0}T_{\mathrm{ch\_1}}\cdot{}^{\mathrm{end}}_{1}T_{\mathrm{ch\_1}}\cdot V \\ &= \begin{bmatrix} \sqrt{l_0^2+l_1^2}\cos(\theta_1+x)(-\cos\theta_0) \\ \sqrt{l_0^2+l_1^2}\cos(\theta_1+x)(-\sin\theta_0) \\ \sqrt{l_0^2+l_1^2}\sin(\theta_1+x) \\ 1 \end{bmatrix}. \end{aligned} \tag{7} $$

Here, ${}^{j}_{j-1}T_{\mathrm{ch\_1}}$ is the transformation matrix from joint $j$ to joint $j-1$ in the work chain 1. $V$ is equal to $\begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix}^T$. Then, we have

$$ \sin(\theta_1 + x) = \frac{{}^{\mathrm{E}}_{\mathrm{S}}Z}{\sqrt{{l_0}^2 + {l_1}^2}}, \tag{8} $$

$$ \sin\theta_0 = \frac{{}^{\mathrm{E}}_{\mathrm{S}}Y}{-\sqrt{{l_0}^2 + {l_1}^2}\cos(\theta_1 + x)}, \tag{9} $$

**TABLE 4** Ranges of joint angles in hand

| Joint | Range (°) | Joint | Range (°) |
|---|---|---|---|
| LShoulderPitch | [–119.5,119.5] | RShoulderPitch | [–119.5,119.5] |
| LShoulderRoll | [–18,76] | RShoulderRoll | [–76,18] |
| LElbowYaw | [–119.5,119.5] | RElbowYaw | [–119.5,119.5] |
| LElbowRoll | [–88.5,–2] | RElbowRoll | [2,88.5] |

$$\cos\theta_0 = \frac{{}^{\mathrm{E}}_{\mathrm{S}}X}{-\sqrt{{l_0}^2 + {l_1}^2}\cos(\theta_1 + x)}. \tag{10}$$

According to (9), the calculated angle lies within the range of [−90, 90]. However, for the joints of the robot, the angle may exceed this range. Table 4 lists the joint-angle ranges of the Nao robot [23]. Here, we use it together with (10) in order to obtain an appropriate value for the joint angle. The detailed calculated procedure is referred to as Algorithm 1. For example, we calculate the value of "LShoulderPitch," that is, $\theta_0$. First, we determine the range of "LShoulderPitch" according to Table 4. If it lies within [0, 180], we can use (10) to obtain the final value. If it lies within [−90, 90], we use (9) to obtain the value. However, if the range is not within these two ranges, for example, [−119.5, 119.5], we need to use (11) to compute the value.

$$\theta_0 = \begin{cases} \cos^{-1}\left(\dfrac{{}^{\mathrm{E}}_{\mathrm{S}}X}{-\sqrt{{l_0}^2+{l_1}^2} * \cos(\theta_1+x)}\right), \sin\theta_0 > 0 \\ -\cos^{-1}\left(\dfrac{{}^{\mathrm{E}}_{\mathrm{S}}X}{-\sqrt{{l_0}^2+{l_1}^2} * \cos(\theta_1+x)}\right), \sin\theta_0 < 0. \end{cases} \tag{11}$$

**Algorithm** 1. Joint-angle computation

1. **input**: the sine and cosine values of $\theta_i$
2. **output**: The angle should be obtained in the robot angle configuration
3. if the range of $\theta_i \in$ [0,180] then
4. $\theta_i = \cos^{-1}\theta_i$
5. return $\theta_i$
6. end if
7. if the range of $\theta_i \in$ [−90,90] then
8. $\theta_i = \sin^{-1}\theta_i$
9. return $\theta_i$
10. end if
11. if $\sin\theta_i > 0$ then
12. $\theta_i = \cos^{-1}\theta_i$
13. else
14. $\theta_i = -\cos^{-1}\theta_i$
15. end if
16. return $\theta_i$

According to $\theta_0$ (with (11)) and $\theta_i$ (with (8)), we have the matrix of ${}^1_0T_{\mathrm{ch\_3}}$ and ${}^2_1T_{\mathrm{ch\_3}}$. For chain 3, between the position of the end joint and the joint angles, there is

$${}^{\mathrm{W}}_{\mathrm{S}}V^4 = {}^1_0T_{\mathrm{ch\_3}} \cdot {}^2_1T_{\mathrm{ch\_3}} \cdot {}^3_2T_{\mathrm{ch\_3}} \cdot {}^{\mathrm{end}}_3T_{\mathrm{ch\_3}} \cdot V \tag{12}$$

which can be written as

$${}^2_1T^{-1}_{\mathrm{ch\_3}} \cdot {}^1_0T^{-1}_{\mathrm{ch\_3}} \cdot {}^{W}_{S}V^4 = {}^3_2T_{\mathrm{ch\_3}} \cdot {}^{\mathrm{end}}_3T_{\mathrm{ch\_3}} \cdot V. \tag{13}$$

To facilitate the calculation, the left side of (13) can be replaced by

$${}^2_1T^{-1}_{\mathrm{ch\_3}} \cdot {}^1_0T^{-1}_{\mathrm{ch\_3}} \cdot {}^{W}_{S}V^4 = \begin{bmatrix} {}^{\mathrm{W}}_{\mathrm{S}}X' \\ {}^{\mathrm{W}}_{\mathrm{S}}Y' \\ {}^{\mathrm{W}}_{\mathrm{S}}Z' \\ 1 \end{bmatrix} = \begin{bmatrix} l_2\cos\theta_2\sin\theta_3 \\ l_2\sin\theta_2\sin\theta_3 \\ l_2\cos\theta_3 + l_1 \\ 1 \end{bmatrix}. \tag{14}$$

Then, we have

$$\sin\theta_2 = \frac{{}^{\mathrm{W}}_{\mathrm{S}}Y'}{l_2 * \sin\theta_3}, \tag{15}$$

$$\cos\theta_2 = \frac{{}^{\mathrm{W}}_{\mathrm{S}}X'}{l_2 * \sin\theta_3}, \tag{16}$$

$$\cos\theta_3 = \frac{{}^{\mathrm{W}}_{\mathrm{S}}Z' - l_1}{l_2}. \tag{17}$$

Using Table 4 and Algorithm 1, $\theta_3$ and $\theta_2$ can be written as

$$\theta_3 = -\cos^{-1}\frac{{}^{\mathrm{W}}_{\mathrm{S}}Z' - l_1}{l_2}, \tag{18}$$

$$\theta_2 = \begin{cases} \cos^{-1}\dfrac{{}^{\mathrm{W}}_{\mathrm{S}}X'}{l_2 * \sin\theta_3}, \sin\theta_2 > 0 \\ -\cos^{-1}\dfrac{{}^{\mathrm{W}}_{\mathrm{S}}X'}{l_2 * \sin\theta_3}, \sin\theta_2 < 0 \end{cases}, \tag{19}$$

Using the same method, we can obtain the corresponding joint angles for the right hand.

## 4.4 | Smoothing and filtering

Although we have obtained the joint angles for the robot control, these angles suffer from the impulse interference and noise, which may result in the jitter. Figure 6 depicts the computed joint angle of "LEblowYaw" during the movements in Figure 13. We note that there are many impulse interferences. Considering the real-time requirements, we chose the filter with low computational complexity.

As the computed angles possess impulse interference and noise, the use of a single filter is inadequate to remove them completely. Here, we remove them by filtering in two phases. During the first phase, we remove the impulse interference using a clipping filter. Figure 7 depicts the "LElbowYaw" angle after the clipping filter. We note that the impulse

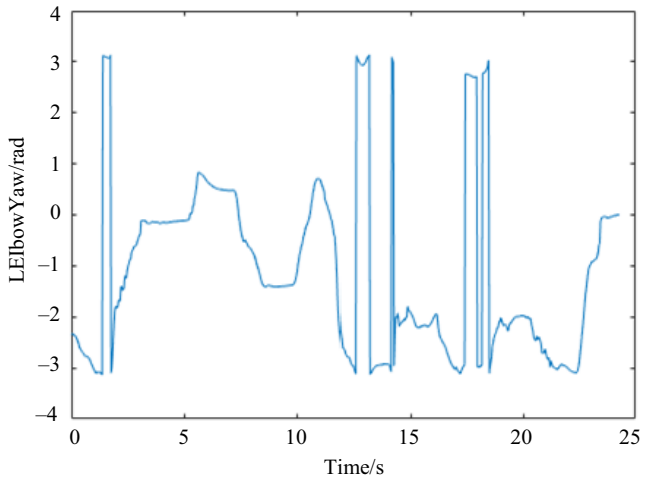


**FIGURE 6** Computed LElbowYaw from Kinect

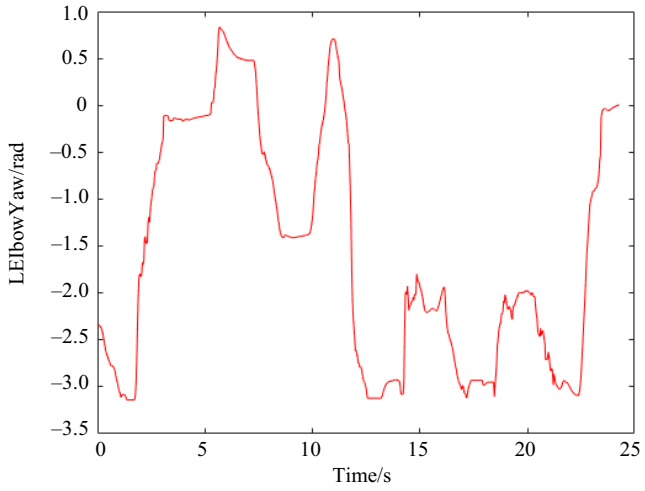


**FIGURE 7** LElbowYaw after a clip filter

interference has been removed completely. However, after the clipping filter, the angle does not change smoothly. Therefore, we used the other filter to remove the noise in the second phase. Figures 8 and 9 depict the angles after the Kalman filter and the five dot-averaging filter. Note that both of these filters can eliminate the noise effectively, but the Kalman filter works better than the five dot-averaging filter. At the same time, we need to compare the hysteresis of both filters. For clarity, Figure 10 illustrates the enlarged signal with these two filters. Note that both of them have a lag of 10 ms–100 ms. The hysteresis is determined by the processed samples and the algorithm itself. Considering that the hysteresis is similar, we chose the Kalman filter as the second phase filter to remove the noise.

## 4.5 | Self-collision avoidance

In order to prevent damage to the robot, self-collision avoidance is necessary. For the Nao robot, the self-collision

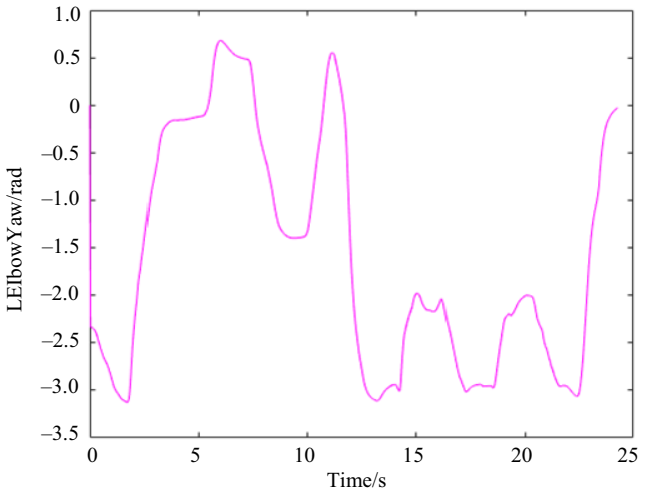


**FIGURE 8** LElbowYaw after Kalman filter

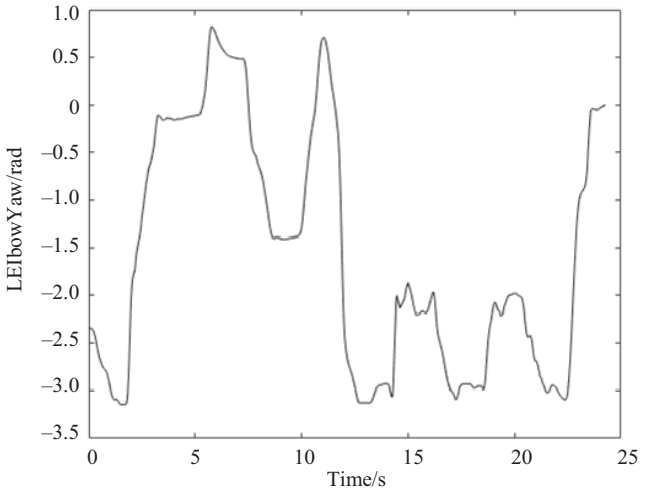


**FIGURE 9** LElbowYaw after the five-point average filter

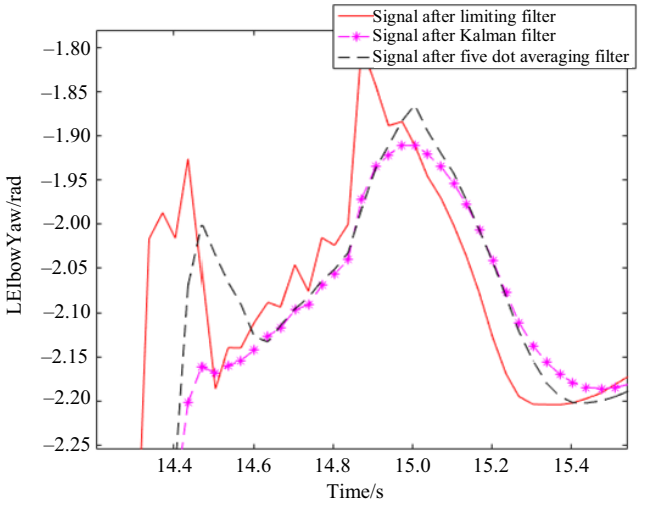


**FIGURE 10** Zoomed out plot of the LElbowYaw

avoidance scheme has been provided by the development platform [24], which uses a 3D shape to simplify the robot structure. By determining whether there is a collision between the 3D shapes, it is possible to know whether the collision will appear.

## 5 | EXPERIMENTAL RESULTS

In our experiment, we used the NAO-H25 robot and Kinect 1.0 to capture human motion. A laptop with an Intel(R) core i5-4460M, 2.5-GHz processor was used to compute the joint angles and control the robot. We compared the motion similarity and delay using the geometric method [5] and the IK method ($IK_{basic}$) [12].

### 5.1 | Similarity of end-effector trajectory

For the similarity analysis of the end-effector trajectory, we performed five motions using the right arm (1–5 in Figure 11) and seven motions with both hands (6–12 in Figure 11). For each motion, we perform 10 iterations to find the average. Table 5 lists the mean-squared error (MSE) in units of centimeters (cm) between the target and the actual end-effector trajectories under different conditions. From the results, we note that the MSE of our solution is on average much smaller than those of other methods. The geometric method performs worst as it configures the robot based on the angle similarity. The $IK_{basic}$ method cannot converge at some points, for example, the singular points and those points beyond the joint angle range of the robot. Besides that, the convergence speed is related to the reference angle computed in the previous frame. Therefore, errors may occur if the movement speed is too fast during the motion imitation.

### 5.2 | Angular similarity

We used the mean cosine similarity between the robot and human upper arm to quantitatively analyze the angular similarity. The calculation vectors consist of four degrees-of-freedom (DoFs) on the robot arm and a corresponding DoF on the human arm. Table 6 lists the average value of the angular similarity. As the geometric method controls the robot with the joint angle, the angular similarity is 100%. In comparison, our $IK_{proposed}$ could achieve a 97.02% angular similarity, which is much better than that of $IK_{basic}$. For clarity, Figure 12 illustrates the imitation results obtained with our solution and the geometric method for different movements. From the results, we find that our solution works well, and is close to those of the geometric method.

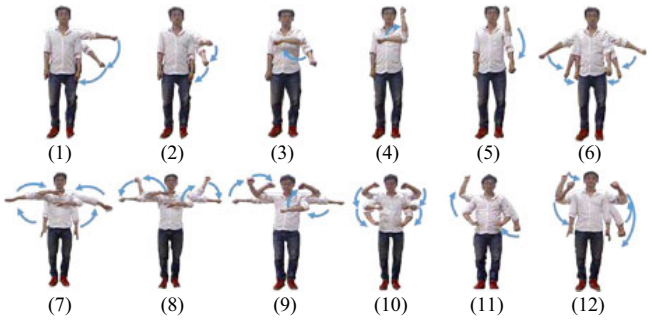


**FIGURE 11** Motions performed with one or two hands

### 5.3 | Mimicking delay

To show the effectiveness of the real-time performance, Table 7 lists the mimicking delay of our system during the

**TABLE 5** Mean squared error (unit: cm)

| Motion no. | $IK_{proposed}$ | $IK_{basic}$ | Geometric |
|---|---|---|---|
| 1 | 0.273 | 2.71E-05 | 13.605 |
| 2 | 0.409 | 9.22E-05 | 5.920 |
| 3 | 0.181 | 9.14E-05 | 10.142 |
| 4 | 1.281 | 19.63 | 8.166 |
| 5 | 0.422 | 10.77 | 13.018 |
| 6 | 0.101 | 2.36E-05 | 9.134 |
| 7 | 1.650 | 4.01E-05 | 14.588 |
| 8 | 1.448 | 15.273 | 13.087 |
| 9 | 1.833 | 12.06 | 14.097 |
| 10 | 1.177 | 6.29E-05 | 13.393 |
| 11 | 1.682 | 1.3E-05 | 12.620 |
| 12 | 1.280 | 13.754 | 13.550 |
| Avg. | 0.978 | 5.957 | 11.777 |

**TABLE 6** Angular similarity between human and robot (%)

| Motion | $IK_{proposed}$ | $IK_{basic}$ | Geometric |
|---|---|---|---|
| 1 | 99.20 | 97.56 | 100.00 |
| 2 | 96.92 | 91.65 | 100.00 |
| 3 | 97.90 | 86.74 | 100.00 |
| 4 | 98.51 | 85.36 | 100.00 |
| 5 | 97.55 | 90.33 | 100.00 |
| 6 | 98.98 | 96.50 | 100.00 |
| 7 | 98.18 | 91.75 | 100.00 |
| 8 | 95.64 | 87.82 | 100.00 |
| 9 | 97.29 | 96.42 | 100.00 |
| 10 | 91.06 | 87.25 | 100.00 |
| 11 | 94.92 | 93.67 | 100.00 |
| 12 | 95.12 | 75.01 | 100.00 |
| Avg. | 97.02 | 90.00 | 100.00 |

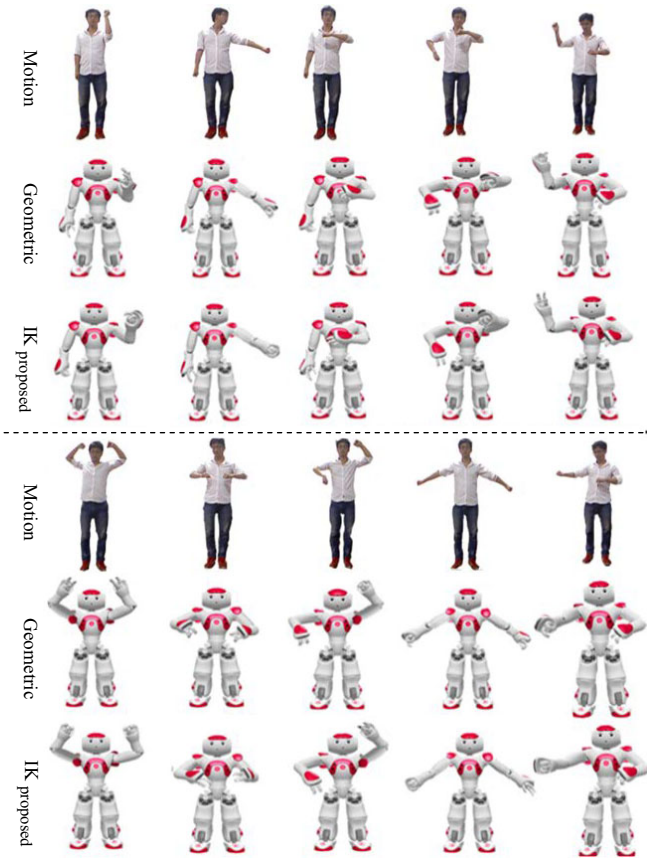


**FIGURE 12** Results of both methods achieved in the same frame for 10 different motions

motions, as shown in Figure 13. However, the delay in Table 7 does not include the response time of the Nao robot. From the results, we note that our proposed system can achieve high real-time performance. The duration of the kinematics solution includes finding the target positions for all end effectors and determining the IK solution to achieve all angles for the robot configuration. The duration of the filtering includes smoothing and filtering. Besides that, the constraint of angles and self-collision avoidance also introduces an extra delay. In our proposed system, the average mimicking delay is about 1.158 ms, and the maximum delay is about 2.357 ms, which is suitable.

## 6 | CONCLUSIONS

In this paper, we proposed a solution to enable the upper-body of a humanoid robot to imitate human motion in real-time. In this solution, we built four work chains on the humanoid robot arms. Then, the IK solution process is simplified by separately solving the IK problem of two work chains on each arm. After that, smoothing and filtering were employed to achieve a safe and reliable imitation. Experimental results verify the effectiveness of our proposed solution in terms of motion similarity and imitation delay. In the future, we intend to extend our solution to the whole body by incorporating balance control for the lower-body of the robot.

**TABLE 7** System delay (unit: ms)

| | Mean | Max |
|---|---|---|
| Kinematics solution | 0.016 | 0.0857 |
| Filtering | 1.142 | 2.336 |
| Total time | 1.158 | 2.375 |

**FIGURE 13** Real-time motion imitation of robot Nao

## ACKNOWLEDGEMENTS

This work was supported by funding from the Key Lab of Broadband Wireless Communication and Sensor Network Technology (Nanjing University of Posts and Telecommunications, Ministry of Education, JZNY201704), Nanjing University of Posts and Telecommunications (NY217021), National Natural Science Foundation of China (Grant No. 61401228), and the Natural Science Foundation of Jiangsu Province (BK20140891).

## ORCID

*Ming Zhang* http://orcid.org/0000-0003-1718-5673
*Jianxin Chen* http://orcid.org/0000-0001-6204-1121

## REFERENCES

1. S. Kim et al., Stable whole-body motion generation for humanoid robots to imitate human motions, *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, St. Louis, MO, USA, Oct. 10–15, 2009, pp. 2518–2524.

2. W. Suleiman et al., On human motion imitation by humanoid robot, *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Pasadena, CA, USA, May 19–23, 2008, pp. 2697–2704.
3. R. Chalodhorn et al., Learning to walk through imitation, *Int. Conf. Artif. Intell. (IJCAI)*, Hyderabad, India, Jan. 2007, pp. 2084–2090.
4. N. Yang et al., A study of the human-robot synchronous control system based on skeletal tracking technology, *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Shenzhen, China, Dec. 12–1, 2013, pp. 2191–2196.
5. I. Almetwally and M. Mallem, Real-time tele-operation and tele-walking of humanoid Robot Nao using Kinect Depth Camera, *Proc. IEEE Int. Conf. Netw., Sens. Contr. (ICNSC)*, Evry, France, Apr. 10–12, 2013, pp. 463–466.
6. F. Zuher and R. Romero, Recognition of human motions for imitation and control of a humanoid Robot, *Conf. Braz. Robot. Symp. Lat. Am. Robot. Symp. (SBR-LARS)*, Fortaleza, Brazil, Nov.29, 2012, pp. 190–195.
7. J. Chen et al., *Lower-body control of humanoid robot NAO via Kinect*, Multimed. Tools Applicat. **77** (2017), no. 9, 10883–10898.
8. J. Koenemann and M. Bennewitz, Whole-body imitation of human motions with a Nao humanoid, *Conf. IEEE Human-Robot Interact. (HRI)*, Boston, MA, USA, Mar. 5–8, 2012, pp. 425–425.
9. J. Koenemann, F. Burget, and M. Bennewitz, Real-time imitation of human whole-body motions by humanoids, *Proc. IEEE Conf. Robot. Autom. (ICRA)*, Hong Kong, China, May 31–June 7, 2014, pp. 2806–2812.
10. F. Wang et al., A real-time human imitation system, *Proc. Intell. Contr. Autom.*, Beijing, China, July 6–8, 2012, pp. 3692–3697.
11. L. Zhang, Z. Cheng, and Y. Gan, Fast human whole body motion imitation algorithm for humanoid robots, *Conf. IEEE Robot. Biomimetics*, Qingdao, China, Dec. 3–7, 2016, pp 1430–1435.
12. S. Mukherjee, D. Paramkusam, and S. K. Dwivedy, Inverse kinematics of a NAO humanoid robot using Kinect to track and imitate human motion, *Conf. RACE*, Chennai, India, Feb. 18-20, 2015, pp. 1–7.
13. M. Alibeigi et al., *Inverse kinematics based human mimicking system using skeletal tracking technology*, J. Intell. Robot. Syst.: Theory Applicat. J. **85** (2017), no. 1, 27–45.
14. Microsoft, *Microsoft Kinect V1.0 information*, Accessed Aug. 14, 2017, available at www.kinectforwindows.com.
15. Xsens, *Xsens MVN motion capture system*, Accessed Aug. 15, 2017, available at www.xsens.com/products/xsens-mvn-animate.
16. B. Krüger et al., A study on perceptual similarity of human motions, *Workshop Virtual Reality Interaction Phys. Simulation*, Lyon, France, Dec. 5–6, 2011, pp. 65–72.
17. J. K. T. Tang et al., *Emulating human perception of motion similarity*, Comput. Animat. Virtual Worlds J. **19** (2008) no. 3–4, 211–221.
18. S. R. Buss and J. S. Kim, *Selectively damped least squares for inverse kinematics*, J. Graph. GPU Game Tools, **10** (2005), no. 3, 37–49.
19. T. F. Chan et al., *A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators*, IEEE Trans. Robot. Autom. **11** (1995), no. 2, 286–292.
20. K. Madsen, H. B. Nielsen, and O. Tingleff, Methods for non-linear least squares problems, *Informatics and Mathematical Modelling*, Technical University of Denmark; Denmark, Apr. 2004, pp. 487–490.
21. N. Kofinas, E. Orfanoudakis, and M. G. Lagoudakis, Complete analytical inverse kinematics for NAO, *Proc. IEEE Int. Conf. Auton. Robot. Syst. (Robotica)*, Lisbon, Portugal, Apr. 24, 2013, pp. 1–6.
22. Aldebaran, *NAO-H25 Joints information*, Accessed Aug. 16, 2017a, available at http://doc.aldebaran.com/21/family/nao_h25/joints_h25.html#h25-joints 2015.
23. J. Angeles, *Fundamentals of robotic mechanical systems*, Springer, USA, 2007, pp. 139–179.
24. Aldebaran, *NAO qi related to self-collision avoidance*, Accessed Aug. 16, 2017b, available at http://doc.aldebaran.com/2-1/naoqi/motion/reflexes-collision-avoidance.html 2015.

## AUTHOR BIOGRAPHIES

**Ming Zhang** is currently studying electronic and communication engineering at the Key Laboratory of Broadband Wireless Communication and Sensor Network Technology (Ministry of Education) at the Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China. His research focus is primarily on human–computer interaction.

**Jianxin Chen** was born in February 1973. He received his Ph.D. degree in electronics engineering from the Shanghai Jiaoton University in 2007. After that, he worked in the Computer College at the Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu, China. From May 2008 to July 2009, He worked as a postdoctoral researcher with the IPP Hurray Research Group, Portugal, and his focus was on real-time human motion tracking with wireless wearable sensor networks. Then, he worked as a researcher at the Spanish research center, Gradiant ETSI Telecommunication, for one year. He is currently an associate professor in the Information and Telecommunication Engineering School at the Nanjing University of Posts and Telecommunications, China. His research interests include cyber-physical systems, body sensor networks, humanoid robots, and human–computer interaction.

**Xin Wei** received his B.S. degree with a major in electronic science and technology from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2005, and his Ph.D. degree in information and communication engineering from the Southeast University, Nanjing, China, in 2009. From 2009 to 2011, he was a postdoctoral researcher with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou, China. Currently, he is an associate professor with the College of Communication and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China. His current research interests include machine learning, pattern recognition, and recommender systems for Internet protocol television (IPTV).

**Dezhou Zhang** is currently studying electronic and communication engineering in the Key Laboratory of Broadband Wireless Communication and Sensor Network Technology (Ministry of Education) at the Nanjing University of Posts and Telecommunications. Currently, his research focuses on human–computer interaction.