# A Human-Following Mobile Robot Providing Natural and Universal Interfaces for Control With Wireless Electronic Devices

Jinfa Chen and Won-jong Kim, *Senior Member, IEEE*

***Abstract*—This article presents the development of a mobile robot using a human-following algorithm based on human path prediction and provides a natural gesture interaction for an operator to control wireless electronic devices using a low-cost red-green-blue-depth (RGB-D) sensor. The overall experimental setup consists of a skid-steered mobile robot, Kinect sensor, laptop, red-green-blue (RGB) camera, and two lamps. OpenNI middleware is used to process the depth data from the Kinect sensor, and OpenCV is used to process data from the RGB camera. The human-following control system consists of two feedback control loops for linear and rotational motions, respectively. A lead-lag and proportional-integral derivative controllers are developed for the linear and rotational motion control loops, respectively. There are small delays (0.3 s for linear motion and 0.2 s for rotational motion) of the system's response. However, the delays are acceptable since they do not cause the tracking distance or angle out of our desirable range ($\pm$0.05 m and $\pm 10^\circ$ of the reference input). A new human-position prediction algorithm based on human orientation is proposed for human following. Experimental results show that the tracking algorithm reduces the distance and angular errors by 40% and 50%, respectively. There are four gestures designed for the operator to control the robot. Success rates of gesture recognition are more than 90% within the detectable range of the Kinect sensor.**

***Index Terms*—Gesture control, human-following mobile robot, human path prediction, human walking modeling.**

## I. Introduction

WITH the development of applications utilizing computer-vision technology and depth cameras, there comes a new way to interact with machines with precision gesture control. One of the most popular applications is the Kinect sensor. The Kinect is a motion sensing device by Microsoft for the Xbox 360 video game console [1]. More recently, smart TVs become equipped with a depth sensor for motion control [2]. However, most of the existing applications use their sensors for only a single or limited number of scenarios. Therefore, in this article, we propose a natural and extendable solution to virtually control electronic devices in the household—a human-following mobile robot with a gesture-based control system to communicate with other electronic devices.

Gesture-based control is regarded a natural interaction interface [3]–[5] because: 1) using a person's own body as an input device eliminates the need for intermediate transducers, and 2) it allows the operator to interact in a way similar to real life. Voice command is another way for human-computer interaction and popular for smart home commercial products, such as Amazon Alexa [6] and Google Home [7]. Compared to voice command, gesture control is superior in the cases that the operator does not know the exact value of the input parameter s/he wants. For example, with a gesture input, an operator can adjust the brightness of a light using a virtual slider in the air to get the brightness s/he wants easily without knowing the exact brightness value. While using voice input, the user has to either try various brightness values or keep increasing or decreasing them.

Low-cost depth cameras [8] have been researched for face [9] and gesture recognition [10], [11], human tracking [12]–[19], and environment mapping [20]. It is ideal for our research because of allowing human skeleton tracking and providing geographic information of the operator. It is an important piece of information for control without requiring additional intervention from the operator. For example, the robot can turn ON the light closest to the operator automatically without given commands. The ideal device should be able to understand the operator's gesture, predict his/her need, and provide help. A Kinect sensor is adopted in this research for gesture recognition and human positioning because it is the most accurate RGB-D sensor in long-range detection in the market [21].

To receive gesture commands through an RGB-D sensor alone, it should be able to keep track of an operator during its operation period. In other words, the mobile robot should be able to follow an operator effectively. To achieve this goal, we developed a feedback control system to maintain a certain distance between the operator and the robot and an anticipation algorithm for human-following.

Many research results on human-following techniques were published in the last two decades. Whereas researchers used to apply laser rangefinders and cameras as sensors for human-tracking [22]–[24], they now tend to use RGB-D cameras

Manuscript received November 30, 2017; revised June 10, 2018 and February 8, 2019; accepted July 7, 2019. Date of publication August 21, 2019; date of current version October 15, 2019. Recommended by Technical Editor G. M. Clayton. *(Corresponding author: Jinfa Chen.)*

J. Chen is with the Huawei Technologies, Santa Clara, CA 95050 USA (e-mail: jchen0213@gmail.com).

W.-J. Kim is with the Department of Mechanical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: wjkim@tamu.edu).

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

Digital Object Identifier 10.1109/TMECH.2019.2936395

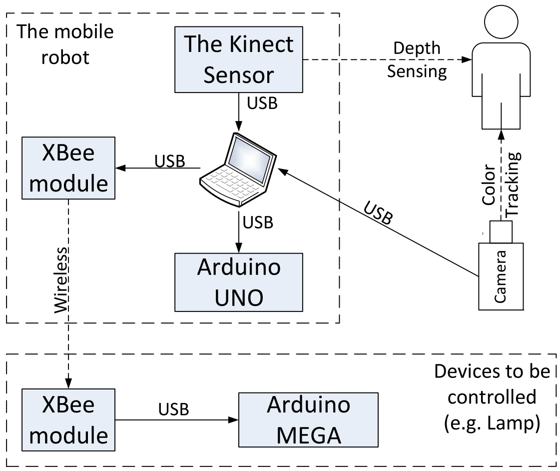


Fig. 1. Overall system architecture.

[13]–[19] or stereo sensors [25], which technically can access human orientation information. However, few such systems utilize this critical information for precise human position prediction. Both Ratsamee, *et al*. [18] and Hu, *et al*. [19] applied the human orientation information to predict the human position based on the assumption that humans always move along their orientation exactly. We agree that human orientation anticipates the moving direction [26], but it does not have to be exactly the same, especially when humans turn or step to the side to avoid an unexpected obstacle. In this article, we propose a new method to predict the human position based on human-orientation data, assuming that the actual motion direction may or may not be the human orientation. We also compare the result with the prediction to that without the prediction and prove that our prediction algorithm improves the tracking performance.

The remainder of this article consists of the following sections. The hardware and software components of the system are described in the second section. Section III covers the human tracking algorithm, a human-walking model for position prediction, and control systems design for the robot to follow a person. In the Section IV is the explanation of the gesture control system. In Section V, experiments are carried out and discussed to evaluate the performances of the human-following system and the gesture control system. The Section VI entails the conclusions of this article.

## II. System Architecture

The overall platform consists of three devices, a mobile robot, RGB camera, and digitally controlled lamp, as shown in Fig. 1. The communication between the mobile robot and the lamp is achieved by a pair of XBee (ZigBee) radio modules. ZigBee modules are wildly used for data transmission in the Internet of Things [27]. It was chosen in this research for its low power consumption, and its data-transmission performance meets our requirements [28] in that the maximum delay is 0.2 s within 5 m when the throughput is 32 kbps. For the mobile robot, the XBee modules, Kinect sensor, Arduino board, and RGB camera are connected to the laptop through four USB cables. For the digital lamp, the other XBee module is connected to the Arduino board.

There are two operation modes–the human-tracking mode and gesture-control mode. In the human-tracking mode, the laptop processes the position data from the Kinect sensor and RGB camera, calculates the distance and angular position errors, plans robot's motion, and then sends the commands to the Arduino board on the robot to track the designated person. For the gesture-control mode, the laptop processes the positional data of joints of the person being tracked by the Kinect sensor and sends pulse width modulation signals to the Arduino board of the lamps through the XBee modules. Lastly, the Arduino board processes these signals to turn the lamp ON and OFF.

## III. Human-Tracking Control

This section contains three parts—human-tracking methodology, human position prediction, and control systems design.

### A. Human-Tracking Methodology

Two human-tracking algorithms are developed for the on-board Kinect sensor and the wide-angle camera mounted on the ceiling, respectively. Data captured by the Kinect sensor are adopted for the feedback control loop whereas the data captured by the RGB camera are used to evaluate the precision of the Kinect sensor only, hence, the robot can fully function without the RGB camera.

*1) Tracking by the Kinect Sensor:* The human detection based on a Kinect sensor relies upon the OpenNI. Since the Kinect sensor is mounted on top of the robot, the reference coordinate keeps changing while following an operator. Therefore, the ego-motion compensation [23] is applied while processing the recorded human position.

Once the spatial coordinates of the operator are obtained, the distance and angle offsets are to be computed in order to run and reorient the mobile robot according to the motion of the operator. In this research, only two coordinates, $x$ and $z$, are considered, based on the Kinect reference coordinate system, as shown in Fig. 2(a). Considering the geometry in Fig. 2, the absolute distance of the operator (neglecting $y$ direction) is calculated as follows:

$$d_a = \left(\sqrt{d_x^2 + d_z^2} - \tan\left(\varphi_h\right)\right)\cos\left(\varphi_h\right) \quad (1)$$

where $d_a$ is the distance to the operator, $\varphi_h$ is the angle between the centerline of the Kinect sensor and the horizontal plane, $d_x$ and $d_z$ indicate the distance of the operator along with the $x$- and $z$-axes. Then the angle between the $z$-axis and the operator $\theta_{err}$ would be

$$\theta_{err} = \text{atan}\left(\frac{d_x}{d_z}\right). \quad (2)$$

*2) Tracking by the RGB Camera:* In this research, the object tracking based on the camera is realized by the color-detection function of OpenCV [29]. To track the operator and the robot by applying the color-tracking algorithm alone, s/he is asked to put on a cap with a specific color (the requirement is for verification only), and the robot is decorated with two different color tapes.

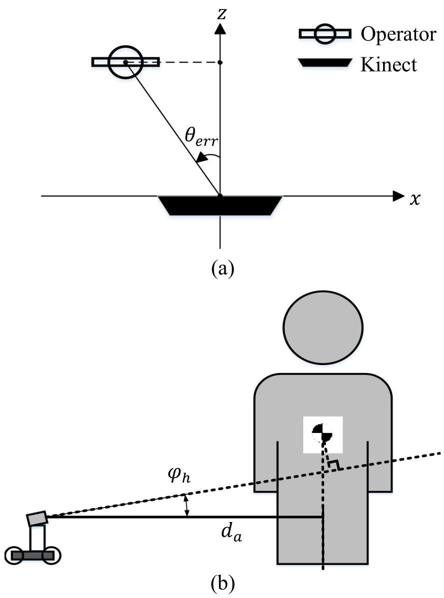


Fig. 2. Definitions of (a) $\theta_{err}$ (top view) and (b) $\varphi_h$ (side view) of an operator in the Kinect's coordinate frame.

All the colors assigned to the operator and the robot are unique, preventing the camera from mistaking other objects.

To calculate the distance between the operator and the robot, the coordinates of the pixels should be transformed to be of a unit of length. Fig. 3 is a schematic diagram to clarify how the camera captures a picture. The width of the view here corresponds to that of the frame. To calculate $\theta_w$, the angle of the view in this direction, the maximum length of the floor that the camera can capture is measured as $L_{fw}$. The height of the camera is measured as $H_f$. Hence, $\theta_w$ can be calculated from

$$\tan\left(\frac{\theta_w}{2}\right) = \frac{L_{fw}}{2H_f}. \tag{3}$$

Assuming the operator is of height $H_t$, the height from the camera to the operator's hat is $L_t = H_f - H_t$. Hence, the absolute distance from the left edge to the operator in the $x$-direction should be

$$l_{xt} = 2L_t \tan\left(\frac{\theta_w}{2}\right)\frac{x_t}{W} \tag{4}$$

where $x_t$ is the distance of the operator to the left edge in pixels, and $W$ is the width of an image captured by the camera in pixels. The actual length of the robot in the $x$-direction can be obtained in a similar way.

$$l_{xr} = 2L_r \tan\left(\frac{\theta_w}{2}\right)\frac{x_r}{W} \tag{5}$$

The height of the robot to the camera, $L_r$, is longer than $L_t$, and the displacement is calculated from the left edge. So to calculate the distance between the operator and the robot in the $x$-direction, $l_{xrt}$ should be known as

$$l_{xrt} = (L_r - L_t)\tan\left(\frac{\theta_w}{2}\right). \tag{6}$$

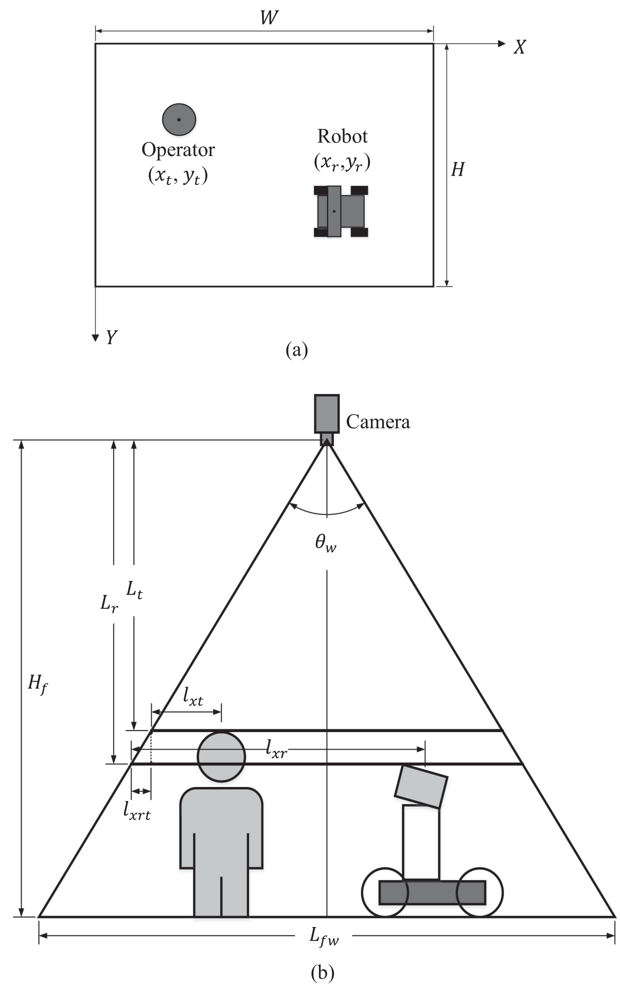


Fig. 3. Schematic of camera-based tracking with (a) top view and (b) front view.

The distance between the operator and the robot in the $x$-direction can be obtained by

$$l_{xa} = l_{xr} - l_{xt} - l_{xrt}. \tag{7}$$

Lastly, the distance between the operator and the robot in two-dimensional form is

$$l_a = \sqrt{l_{xa}^2 + l_{ya}^2} \tag{8}$$

where $l_{ya}$ is the distance between the operator and the robot in the $y$-direction.

## B. Human Position Prediction

In this section, a new algorithm to predict the operator position based on the current position and orientation is proposed to allow for smoother and faster human-following. Fig. 4 is a schematic to illustrate the human-walking model for position prediction.

The state vector of the operator is modeled as

$$\boldsymbol{U_k} = \left[\, x_k \; y_k \; o_k \; dl_k \; dr_k \,\right]^T \tag{9}$$

where $(x_k,\ y_k)$ are the $x$- and $y$-coordinates of the operator in the world coordinate frame; according to the frame index, $k$, $o_k$

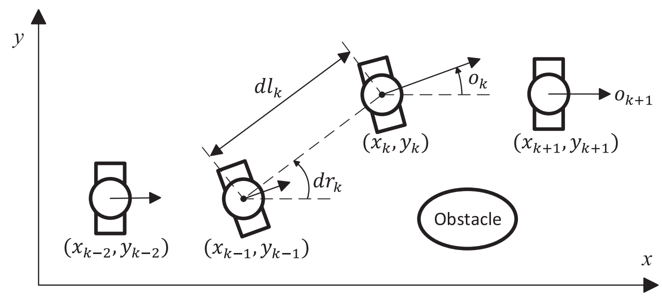


Fig. 4. Human-walking model for prediction.

is the body orientation, $dl_k$ represents the displacement of the operator, and $dr_k$ is the direction of the displacement from the last frame to the current frame. A human walking model for operator position prediction is given by

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + dl_{k+1}\cos(dr_k) \\ y_k + dl_{k+1}\sin(dr_k) \end{bmatrix}. \tag{10}$$

The difference of the body orientation and the actual moving direction is

$$\varphi_k = dr_k - o_{k-1}. \tag{11}$$

Assuming that the orientation of a human anticipates the direction of his/her displacement [26], and the difference between $o_{k-1}$ and $dr_k$ varies linearly; the rate of the linear displacement is also the same. $dr_{k+1}$ and $dl_{k+1}$ can be derived as

$$\begin{aligned} dr_{k+n} &= o_k + 2\varphi_k - \varphi_{k-n} \\ dl_{k+n} &= 2dl_k - dl_{k-n} \end{aligned} \tag{12}$$

where $n$ is the time step for prediction. The sampling period of the human-tracking system $T$ is set to 1/30 s because the maximum frame rate of the Kinect sensor is 30 frame per second and shorter $T$ does not improve the performance significantly. The prediction time period in the future, $nT$, is a critical parameter for human-position prediction. In other words, this is the time required to predict the position of the operator. If this time period is too short, the predicted position will be very close to the current position. Otherwise, the prediction will have a large error because the algorithm runs for a short-term prediction only. Therefore, selecting a proper prediction time period for human-position prediction is important. Experimental results show that the proper time period to update the predicted position is 0.2 s for our system.

As shown in Fig. 5, the trajectory of the operator in the figure includes four cases of tracking:

1) U0-U1: The operator walks straight along with his/her body direction.
2) U1-U2: The operator begins to turn left, so his/her body directs to the left side abruptly. Since the body's direction should be tangent to the actual trajectory, there is an angle between the displacement U1-U2 and the body direction of the operator in U1.
3) U2-U3: The operator keeps turning left, and there is still an angle between the actual trajectory and the body direction.

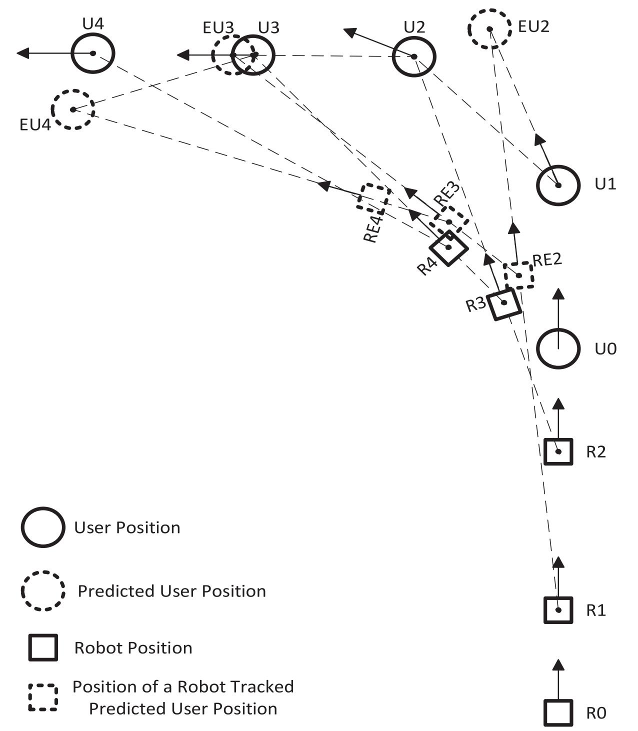


Fig. 5. Schematic of the tracking algorithm.

4) U3-U4: The operator stops turning and keeps walking forward again. The actual displacement is along with the body direction.

The goal of human tracking in this article is to maintain a certain distance and angle between the operator and the robot. Fig. 5 demonstrates the working principle of the algorithm to predict and track the actual position of the operator.

*1) Tracking the Actual Motion of the Operator:* In the first frame, the operator and the robot are at U0 and R0, respectively. After detecting the operator at U0, the robot will move to R1. In the meantime, the operator moves to U1. Then the robot tracks U1, and keeps following the new position of the operator in every frame, and there are R2, R3, etc.

*2) Tracking the Predicted Operator Position:* Since the operator is at the initial position, there is no information of the last movement, the robot tracks U0 and arrives at R1. In the second frame, the predicted position of the operator is at U1, EU2 is calculated and the robot goes tracking EU2 instead of U1 and arrives at RE2.

## C. Control Systems Design

The mobile robot used in this research is a skid-steered one. There are several assumptions to develop its mathematical model.

1) The vehicle is symmetric about its longitudinal and latitudinal axes.
2) Its center of mass is at its geometric center.
3) There is point contact between the wheels and the ground.
4) Wheels on the same side have the same angular velocity.
5) The robot runs on a flat surface, and its four wheels are always in contact with the surface.

TABLE I
SUMMARY OF THE KEY PARAMETERS AND VARIABLES OF THE ROBOT

| | |
|---|---|
| Vehicle width | $w_v$ |
| Vehicle length | $d_v$ |
| Terrain-dependent parameter | $\alpha$ |
| Vehicle displacement in longitudinal direction | $y$ |
| Vehicle rotation | $\varphi$ |
| Vehicle velocity in longitudinal direction | $V_y$ |
| Vehicle angular velocity | $\dot{\varphi}$ |
| Rotation angle of the left wheel | $\theta_L$ |
| Rotation angle of the right wheel | $\theta_R$ |
| Angular velocity of wheels on the left side | $\omega_L$ |
| Angular velocity of wheels on the right side | $\omega_R$ |
| Input torque of the left motors | $T_L$ |
| Input torque of the right motors | $T_R$ |

Table I lists all the critical parameters and variables of the robot. For vehicles that satisfy the previous assumptions, an experimental kinematic model of a skid-steered vehicle developed in [30] is given by

$$\begin{bmatrix} V_y \\ \dot{\varphi} \end{bmatrix} = \frac{r}{\alpha w_v} \begin{bmatrix} \frac{\alpha w_v}{2} & \frac{\alpha w_v}{2} \\ -1 & 1 \end{bmatrix} \begin{bmatrix} \omega_L \\ \omega_R \end{bmatrix}, \tag{13}$$

and the dynamic model is given by

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{G}(\boldsymbol{q}) = \boldsymbol{\tau} \tag{14}$$

where $\boldsymbol{q} = [\theta_L \; \theta_R]^T$ is the angular displacement of the left and right wheels, $\dot{\boldsymbol{q}} = [\omega_L \; \omega_R]^T$ is the angular velocity of the left and right motors, $\boldsymbol{M}$ is the mass matrix, $\boldsymbol{G}(\boldsymbol{q})$ is the gravitational term. Since only planar motion is considered, $\boldsymbol{G}(\boldsymbol{q}) = 0$. $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ denotes the resistance resulting from the interaction between the wheels and terrain. Based on the analysis of resistance term in [30] and considering that the model is only used to derive the feedback controller, the resistance term $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is simplified to a linear term

$$\boldsymbol{C}(\dot{\boldsymbol{q}}) = \begin{bmatrix} r(\mu_r + \mu_f) & -r\mu_f \\ -r\mu_f & r(\mu_r + \mu_f) \end{bmatrix} \dot{\boldsymbol{q}} \tag{15}$$

assuming that the resistance is proportional to the angular velocity. $\mu_r$ and $\mu_f$ are the rolling resistance coefficient and the friction coefficient, respectively. Therefore, (14) becomes

$$\begin{bmatrix} \frac{mr^2}{4} + \frac{r^2 I}{\alpha {w_v}^2} & \frac{mr^2}{4} - \frac{r^2 I}{\alpha {w_v}^2} \\ \frac{mr^2}{4} - \frac{r^2 I}{\alpha {w_v}^2} & \frac{mr^2}{4} + \frac{r^2 I}{\alpha {w_v}^2} \end{bmatrix} \ddot{\boldsymbol{q}} + \begin{bmatrix} r(\mu_r + \mu_f) & -r\mu_f \\ -r\mu_f & r(\mu_r + \mu_f) \end{bmatrix} \dot{\boldsymbol{q}} = \begin{bmatrix} \tau_L \\ \tau_R \end{bmatrix}. \tag{16}$$

Then, the state-space matrix form can be deduced, and the transfer functions will be

$$\begin{bmatrix} Y_y \\ Y_\varphi \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} T_L \\ T_R \end{bmatrix}, \tag{17}$$

where $G_{11}(s) = G_{12}(s)$ and $G_{22}(s) = -G_{21}(s)$. Assuming the linear motion and the rotational motion are independent, the multiple-input-multiple-output (MIMO) system can be decoupled into two single-input-single-output (SISO) systems for simpler analysis. When there is linear displacement only, the angular displacement output $Y_\varphi$ is zero, torque $T_L$ equals torque $T_R$, the displacement output can be derived as following:

$$Y_y = G_{11}(s)\, T_L + G_{12}(s)\, T_R = 2G_{11}(s)\, T_L. \tag{18}$$

Therefore, the plant $G_y(s)$ for linear motion only will be

$$G_y(s) = \frac{Y_y}{T_L} = 2G_{11}(s). \tag{19}$$

Similarly, plant $G_\varphi(s)$ for linear motion only will be

$$G_\varphi(s) = \frac{Y_\varphi}{T_R} = 2G_{22}(s). \tag{20}$$

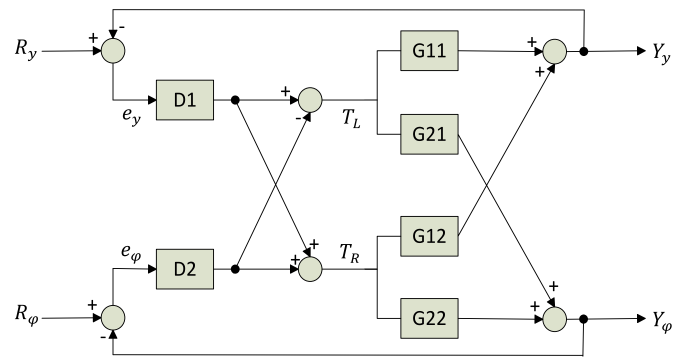


Fig. 6. Block diagram of the control system.

The block diagram of the MIMO control system of the mobile robot is given in Fig. 6. $R_y$ and $R_\varphi$ are the reference inputs of displacement and rotation, respectively, and $e_y$ and $e_\varphi$ are the errors of the linear and rotational motion, respectively. $D_1(s)$ and $D_2(s)$ are the lead-lag controller for linear motion and the lead controller for rotational motion, respectively. Four transfer functions $G_{ij}(s)$ represent the plants derived from the mathematical model. $Y_y$ and $Y_\varphi$ are the translation and rotation outputs, respectively.

Two proportional-integral derivative (PID) controllers were originally designed and tuned with the Matlab SISOtool for both linear and rotational motion control. Since there was a 0.05-m steady-state error, a lead-lag controller was developed for linear motion control to eliminate the steady-state error. Equation (21) is the lead-lag controller for linear motion, and (22) is the PID controller for rotational motion

$$D_1(s)_{lead-lag} = 1.25 \frac{(s + 2.85)(s + 0.29)}{(s + 4.07)(s + 0.28)} \tag{21}$$

$$D_2(s)_{PID} = 0.862 + \frac{0.02}{s} + 0.0086s. \tag{22}$$

## IV. GESTURE CONTROL

In this research, the Kinect sensor is used to track all the joints of an operator's body, and OpenNI is used as a middleware to access these data. To achieve a practical gesture-based interaction system, a gesture-control system was improved and extended, and four gestures were defined, aiming to implement a vision-

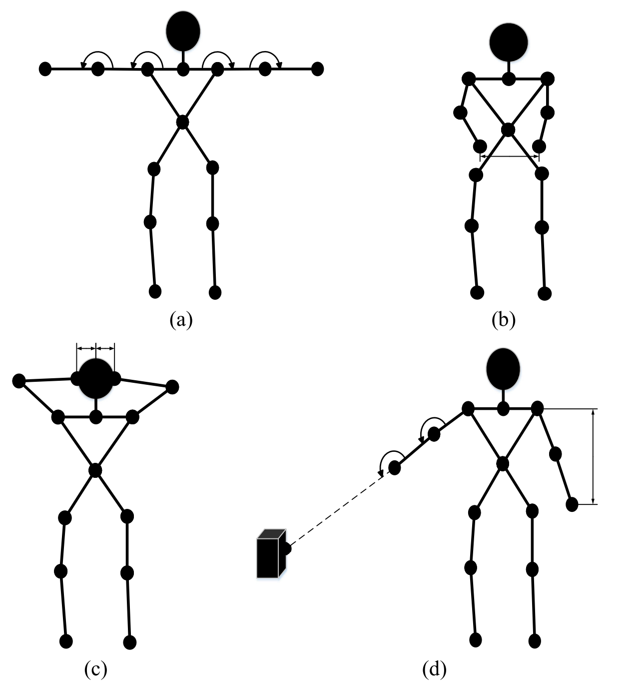


Fig. 7. Gesture (a) to switch to the controlling mode, (b) to switch to the tracking mode, (c) to create a virtual lamp, and (d) to select a lamp and adjust its color.

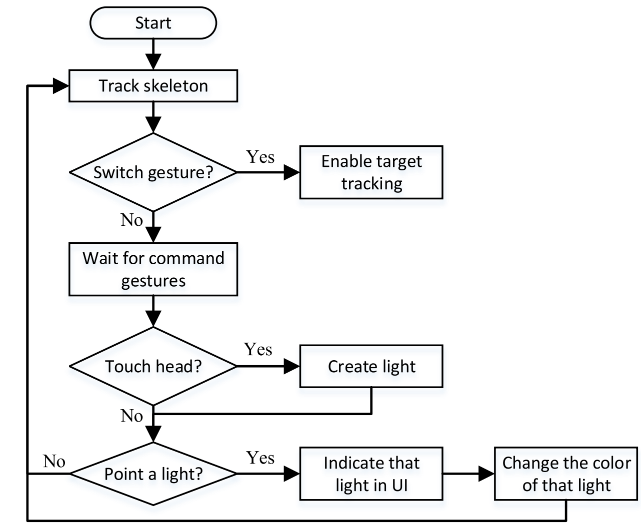


Fig. 8. Flowchart of the gesture-control mode.

based gesture recognizer [31]. Within the whole set of defined gestures, the system is designed to recognize only the body.

### A. Gesture Definition

The vocabulary consists of the four gestures presented below:

1) Switch to controlling;
2) Switch to tracking;
3) Lamp creation;
4) Lamp selection and color control.

The gestures are defined by stipulating certain angles of joints. The four gestures are just designed to show the robot's interaction capability; it is easy to increase the number of recognizable gestures by adding definitions.

Gesture 1: Switch to controlling

This gesture is used to switch from the tracking mode to the gesture-controlling mode. It requires the operator to raise his/her arms laterally. This gesture is chosen for three reasons. First, it is easy to execute. Second, it is easy to be recognized by the Kinect because there is no overlap of the body. Third, normally humans do not raise their arms that high while they walk, so it is hard to be mistaken. To inform the robot that the operator is performing this switch gesture, four angles of joints are stipulated to be close to 180° (±10°), as shown in Fig. 7(a). Therefore, whenever all these angles meet the requirements of the stipulation, actions of this mode-switch gesture will be triggered.

Gesture 2: Switch to tracking

This gesture is used to switch from the gesture controlling mode to the tracking mode. The gesture requires the operator to expose his/her arms in front of the Kinect sensor and makes them closer. To inform the robot that the operator is performing this switch gesture, the distance between the operator's hands must be closer than 0.01 m as shown in Fig. 7(b). Therefore, whenever the distance meets the requirement of the stipulation, actions of this mode-switch gesture are triggered.

Gesture 3: Lamp creation

Gesture in Fig. 7(c) is to create a virtual lamp in the reference coordinate system of the Kinect. Once this gesture is recognized, the position of the operator's center of mass will be recorded and regarded as the position of the lamp, a rectangle will be drawn in the graphical user interface (GUI) window to indicate the location of the lamp.

This gesture requires the operator to touch his/her head with hands and is defined by calculating the distance between each hand and the head of the operator. If the distance is around 0.1 m (±0.05 m), the lamp-creation gesture will be triggered.

Gesture 4: Lamp selection and color control

This gesture is used to tell the robot which lamp the operator selected and to change the color of the lamp. When the system recognizes the operator points to one of the lamps created before, the rectangular box in the GUI will turn white to indicate that the lamp was selected, as shown in Fig. 7(d). In the meantime, the color of that lamp will change according to the height of the operator's left hand.

### B. Gesture Control

For the gesture-control system, as shown in Fig. 8, in the beginning, the system tracks the operator's skeleton. Then the system determines whether s/he is performing the gesture for switching to the gesture-control mode or not. If yes, it switches to the human-tracking mode. If no, it evaluates specific joint angles and joint distances to decide if the operator is touching his/her head or pointing to a lamp.

## V. Experimental Results

In this section, the performance of the controllers are evaluated first, then the human position prediction algorithm and the gesture recognition system are described.

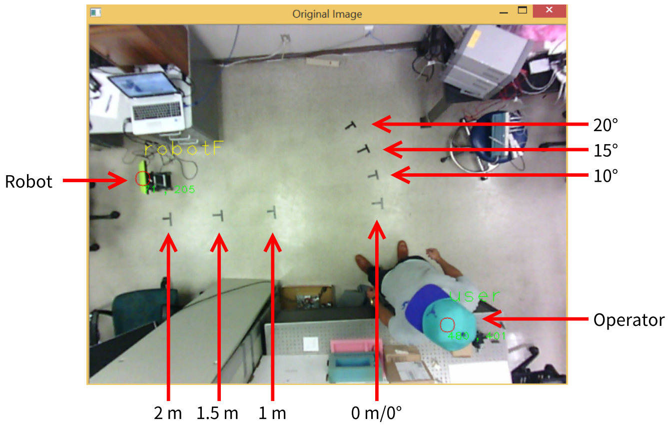


Fig. 9. Configuration of the experimented platform.

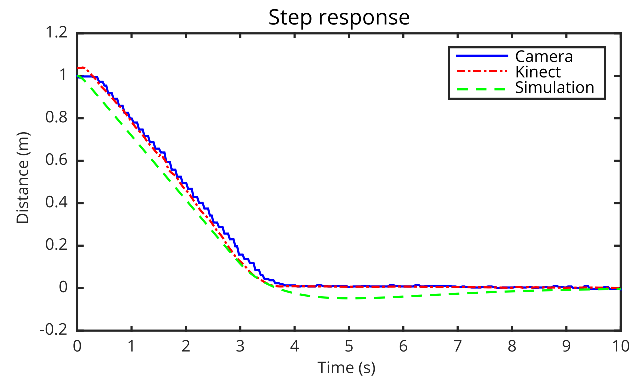


Fig. 10. Step responses of the distance-control loop.

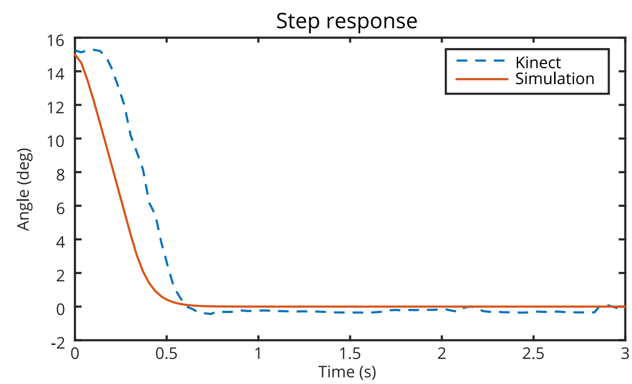


Fig. 11. Step responses of the angle-control loop.

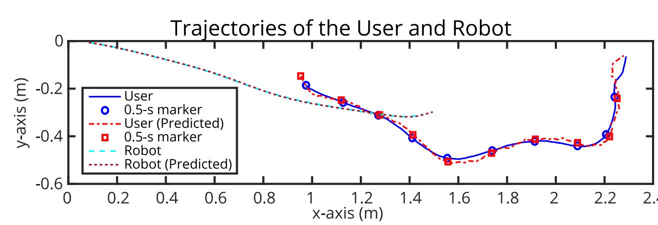


Fig. 12. Trajectories of an operator and the robot.

## A. Control Systems Evaluation

The experimental platform is shown in Fig. 9. The robot is to the left and the operator is at the bottom right. There are seven black marks in a "T" shape on the floor to indicate the distance and angle between the operator and the robot. The mark nearest to the operator is the "0 m/ 0°" mark. The three marks to the left denote the distances (from left to right are 2, 1.5, and 1 m, respectively). The other three marks in the top middle denote the angles (from top to bottom are 20°, 15°, and 10°, respectively). For linear motion testing, the operator always stands at the "0 m/ 0°" mark while the robot starts at one of the three distance marks to the left. For rotational motion testing, the robot always starts at the "2 m" mark while the operator stands at one of the degree marks to the right.

In this experiment, two sets of tests were carried out to obtain the linear and rotational motion response data of the lead-lag and PID controllers, respectively. The robot is placed on the 2-m mark. The operator stands still in front of the robot on the 10°, 15°, or 20° marks. During the experiment, the distance and the angle between the robot and the operator detected by both the Kinect sensor and the camera are recorded. The angle reference input is set to be 0°, and the distance reference input is set to be 1 m. Therefore, when the tracking begins, the robot runs and turns to the operator simultaneously.

Fig. 10 shows the step responses of the distance-control loop with the initial angle of 15°. The solid line represents the result of using the RGB camera, and the dash-dotted line, the result of using the Kinect sensor. The two lines match well, which implies the distance error estimated is very close to the actual one. The experimental result also matches the simulation well with a small delay about 0.3 s with the error reduced to 0 after 3 s. Fig. 11 shows the step responses of the angle-control loop. The experimental result matches the simulation result well, with a delay about 0.2 s at maximum and small oscillations. Although the results in this set of experiments do not fit the simulation result perfectly, tests show that these delays and steady-state errors are acceptable.

Additional experiments were carried out to determine the actual range of the distance and the angle between the robot and the operator when the robot follows an operator walking at a normal speed [26]. They were about 0.8–1.8 m and ±18°, which are within the detectable range 0.4–4 m and ±23.5° [21]. Hence, the delays and steady-state errors are acceptable, and they do not drive the robot out of the controllable ranges.

## B. Evaluation of the Prediction Algorithm

In Fig. 12, the solid line to the right is the operator's trajectory, and the dash-dotted line is the trajectory of the predicted operator position. The dashed line to the left is the trajectory of the robot, which followed the current position of the operator, and the dotted line is the trajectory of the robot, when followed the predicted position of the operator. As shown in Fig. 12, the operator position was predicted precisely by the new algorithm as the dash-dotted line matches the solid line well. Some errors occur after the operator accelerates the turning rate (case 2, as

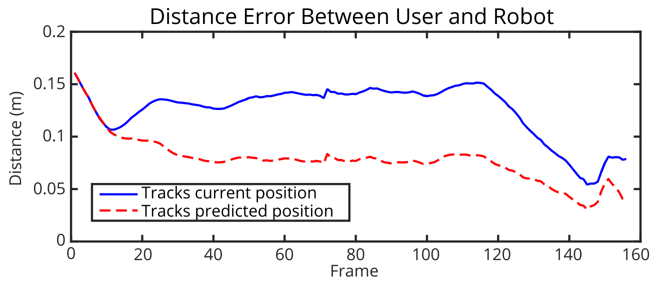


Fig. 13. Distance error by applying the two tracking algorithm.

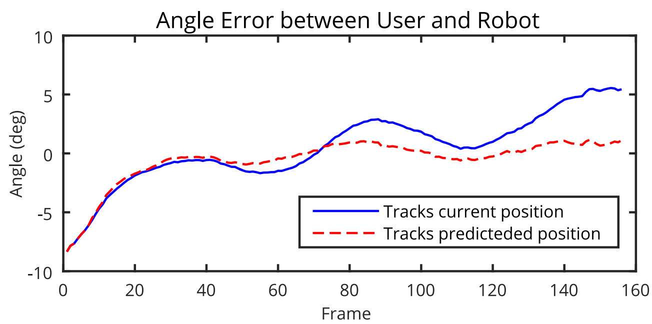


Fig. 14. Angular error by applying the two tracking algorithms.

analyzed in the prediction algorithm section). However, the error is small about 0.05 m, and the dotted line perfectly matches the dashed line. This implies that the robot follows the same trajectory and arrives at the ideal location sooner because the predicted position is obtained 0.2 s ahead of the real one.

Fig. 13 shows the distance error (absolute distance between the operator and robot minus the desired distance) of the robot tracking the operator directly and tracking the predicted position, respectively. As shown in the figure, the error was reduced by about 40% after utilizing the new tracking algorithm.

Fig. 14 shows the angular error between the operator and the robot using the direct tracking method and the new tracking method. Experimental results indicate that the angular error was reduced by more than 50%.

### C. Gesture Recognition Evaluation

Experiments were performed to assess the reliability of the gesture-recognition subsystem through the assessment of the recognition success rate on the five gestures defined in the previous section–two mode-switch gestures, lamp-creation, lamp-selection, and color change gesture. For these tests, ten people, aged 26 years on average, were asked to perform all the gestures that the robot can recognize, and the results are tabulated in Table II.

Table II shows the average success rates of the individual testing from 1.5 or 2 m away from the robot. For the test from 1.5 m away, every gesture was recognized well at 0° and 10°. When the test went to 15°, the success rates of gestures decreased. When it came to 20°, the operator was out of the Kinect's detectable angular range. For the test from 2 m away, every gesture was

TABLE II
SUCCESS RATE OF THE GESTURE RECOGNITION AS DESCRIBED IN FIG. 7

| | 1.5 m | | | 2 m | | | |
|---|---|---|---|---|---|---|---|
| | 0° | 10° | 15° | 0° | 10° | 15° | 20° |
| Switch 1 | 100% | 100% | 83% | 100% | 100% | 93% | 83% |
| Switch 2 | 100% | 100% | 90% | 100% | 100% | 100% | 93% |
| Create lamp | 100% | 100% | 87% | 100% | 100% | 97% | 90% |
| Select lamp | 100% | 100% | 80% | 100% | 100% | 93% | 83% |
| Change color | 100% | 100% | 93% | 100% | 100% | 100% | 93% |

recognized well at 0° and 10°. As the angle increased from 10° to 20°, the success rates of all the gestures decreased. The reason for the success-rate decrease was that the Kinect could not track the crucial part of the body that triggered the motion. In other words, the critical body part was out of its best detectable range. According to these data, when the robot is serving at a distance of 1.5 to 2 m, the operator was not supposed to have an angle greater than 20° from the line of sight to the robot.

## VI. CONCLUSION

The affordable and reliable robotic system developed in this article achieved the goal of demonstrating the potential for wide usage of its natural interface. First, the robot showed its ability to interact with other electronic devices wirelessly. The robot's human-tracking ability was evaluated and demonstrated that it was reliable within the detectable range of the Kinect. Therefore, with a wireless communication module, the robot can provide a natural interface allowing control of a variety of electronic devices. In addition, a novel human-tracking algorithm based on human position prediction was proposed in this article and verified successfully, reducing tracking errors. The prediction models in [18], [19] were based on the assumption that the actual moving direction was the same with the human orientation, whereas our model assuming that the actual moving direction is related to the human orientation, but may or may not match the human orientation precisely. Moreover, they did not clearly explain how their prediction algorithms were implemented for human following. For our prediction algorithm, the best prediction time was 0.2 s, and the prediction error was within 0.05 m. We also compared the results with the prediction to that without prediction and proved that the distance and angular errors were reduced by 40% and 50%, respectively.

In the future, the pan-tilt motor in the Kinect sensor can be utilized for a better view of the operator so that it can get a better gesture recognition. Second, some computer-vision frameworks may be utilized to implement object recognition, which will facilitate the human-robot interaction. Third, it would be interesting to integrate additional human-oriented perception systems, for example, speech recognition. In this case, one could take advantage of the hardware already installed on the robot, the array of microphones of the Kinect.

## References


[1] Kinect for Xbox One. 2018. [Online]. Available: https://www.xbox.com/en-US/xbox-one/accessories/kinect
[2] What is Gesture Control in Samsung F Series SMART TV?. Mar. 21, 2018. [Online]. Available: https://www.samsung.com/in/support/tv-audio-video/what-is-gesture-control-in-samsung-f-series-smart-tv/
[3] F. Alemuda and F. J. Lin, "Gesture-based control in a smart home environment," in *Proc. IEEE Int. Conf. Internet Things (iThings) IEEE Green Comput. Commun. IEEE Cyber, Phys. Soc. Comput. IEEE Smart Data*, Exeter, U.K., Jun. 2017, pp. 784–791.
[4] Z. Liang, X. Xu, and C. Qing, "Natural interaction in freehand pointing: Designing the mapping of interaction space," *IEEE Consum. Electron. Mag.*, vol. 6, no. 2, pp. 89–93, Apr. 2017.
[5] V. Villani, L. Sabattini, G. Riggio, C. Secchi, M. Minelli, and C. Fantuzzi, "A natural infrastructure-less human–robot interaction system," *IEEE Robot. Autom. Lett.*, vol. 2, no. 3, pp. 1640–1647, Jul. 2017.
[6] Amazon Echo, 2018. [Online]. Available: http://www.amazon.com/echo
[7] Google Home, 2018. [Online]. Available: https://store.google.com/product/google_home
[8] K. D. Mankoff and T. A. Russo, "The Kinect: A low-cost, high-resolution, short-range 3D camera," *Earth Surface Processes Landforms*, vol. 38, no. 9, pp. 926–936, Jul. 2013.
[9] M. R. Lee, Y. Li, and M. Chien, "Real-time face tracking and recognition using the mobile robots," *Adv. Robot.*, vol. 29, no. 3, pp. 187–208, Feb. 2015.
[10] E. Chatzidaki and M. Xenos, "A case study on learning through natural ways of interaction," in *Proc. IEEE Global Eng. Educ. Conf.*, Athens, Greece, Apr. 2017, pp. 746–753.
[11] J. Cui, A. Kuijper, and A. Sourin, "Exploration of natural free-hand interaction for shape modeling using leap motion controller," in *Proc. Int. Conf. Cyberworlds*, Chongqing, China, Sep. 2016, pp. 41–48.
[12] S. Jia, S. Wang, L. Wang, and X. Li, "Human tracking system based on adaptive multi-feature mean-shift for robot under the double-layer locating mechanism," *Adv. Robot.*, vol. 28, no. 24, pp. 1653–1664, Dec. 2014.
[13] M. Munaro and E. Menegatti, "Fast RGB-D people tracking for service robots," *Auton. Robots*, vol. 37, no. 3, pp. 227–242, Feb. 2014.
[14] C. W. Chen, S. P. Tseng, Y. T. Hsu, and J. F. Wang, "Design and implementation of human following for separable omnidirectional mobile system of smart home robot," in *Proc. Int. Conf. Orange Technol.*, Singapore, pp. 210–213, Dec. 2017.
[15] Y. Sun, L. Sun, and J. Liu, "Real-time and fast RGB-D based people detection and tracking for service robots," in *Proc. 12th World Congr. Intell. Control Autom.*, Guilin, China, Jun. 2016, pp. 1514–1519.
[16] H. Zhang, R. J. Yan, W. S. Zhou, and L. Sheng, "Binocular vision sensor (Kinect)-based pedestrian following mobile robot," *Appl. Mechanics Materials*, vol. 670–671, pp. 1326–1329, Oct. 2014.
[17] M. Shimoyama, N. Matsuhira, and K. Suzuki, "Human characterization by a following robot using a depth sensor," in *Proc. IEEE Int. Symp. Syst. Integr.*, Taipei, Taiwan, Dec. 2017, pp. 998–1003.
[18] P. Ratsamee, Y. Mae, K. Ohara, T. Takubo, and T. Arai, "People tracking with body pose estimation for human path prediction," in *Proc. IEEE Int. Conf. Mechatronics Automat.*, Chengdu, China, Aug. 2012, pp. 1915–1920.
[19] J. Hu, J. Wang, and D. M. Ho, "Design of sensing system and anticipative behavior for human following of mobile robots," *IEEE Trans. Ind. Electron.*, vol. 61, no. 4, pp. 1916–1927, Apr. 2014.
[20] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox, "RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments," *Int. J. Robot. Res.*, vol. 31, no. 5, pp. 647–663, Apr. 2012.
[21] M. G. Diaz, F. Tombari, P. Rodriguez-Gonzalvez, and D. Gonzalez-Aguilera, "Analysis and evaluation between the first and the second generation of RGB-D sensors," *IEEE Sensors J.*, vol. 15, no. 11, pp. 6507–6516, Nov. 2015.
[22] S. Muramatsu, H. Matsuda, S. Nakamoto, D. Chugo, T. Suehiro, and H. Hashimoto, "Consideration of human movement prediction for human friendly mobile robot's path planning," in *Proc. 4th IEEE Int. Conf. Cogn. Infocommunications*, Budapest, Hungary, Dec. 2013, pp. 235–240.
[23] B. Jung and G. S. Sukhatme, "Real-time motion tracking from a mobile robot," *Int. J. Social Robot.*, vol. 2, no. 1, pp. 63–78, Mar. 2010.
[24] V. Alvarez-Santos, X. M. Pardo, R. Iglesias, A. Canedo-Rodriguez, and C. V. Regueiro, "Feature analysis for human recognition and discrimination: Application to a person-following behaviour in a mobile robot," *Robot. Auton. Syst.*, vol. 60, no. 8, pp. 1021–1036, Aug. 2012.
[25] B. Ali, Y. Ayaz, M. Jamil, S. O. Gilani, and N. Muhammad, "Improved method for stereo vision-based human detection for a mobile robot following a target person," *South African J. Ind. Eng.*, vol. 26, no. 1, pp. 102–119, May 2015.
[26] G. Arechavaleta, J. P. Laumond, H. Hicheur, and A. Berthoz, "On the nonholonomic nature of human locomotion," *Auton. Robots*, vol. 25, no. 1, pp. 25–35, Aug. 2008.
[27] M. A. Rashid and X. Han, "Gesture control of ZigBee connected smart home Internet of Things," in *Proc. 5th Int. Conf. Inform., Electron. Vision*, Dhaka, Bangladesh, May 2016, pp. 667–670.
[28] S. Soijoyo and A. Ashari, "Analysis of ZigBee data transmission on wireless sensor network topology," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 9, pp. 145–151, Sep. 2017.
[29] OpenCV, 2018. [Online]. Available: http://opencv.org/
[30] W. Yu, O. Y. Chuy. Jr., E. G. Collins. Jr., and P. Hollis, "Analysis and experimental verification for dynamic modeling of a skid-steered wheeled vehicle," *IEEE Trans. Robot.*, vol. 26, no. 2, pp. 340–353, Apr. 2010.
[31] E. R. Melgar and C. C. Díez, "Mood Lamps," in *Arduino and Kinect Projects: Design, Build, Blow their Minds*. New York, NY, USA: Apress, 2012, ch. 7, pp. 133–165.



**Jinfa Chen** received the B.Eng. degree in machine design & manufacturing and their automation from Shenzhen University, Shenzhen, China, in 2012, and the M.S. degree in mechanical engineering from Texas A&M University (TAMU), College Station, TX, USA, in 2015.

He worked as a Graduate Research Assistant with Texas A&M University, in 2016. Since 2016, he has been a Robotics Software Engineer with Huawei Technologies Inc., Santa Clara, CA, USA. His research interests include robotics control, human following, and advanced robotics.

**Won-jong Kim** (S'89–M'97–SM'03) received the B.S. (*summa cum laude*) and M.S. degrees in control and instrumentation engineering from Seoul National University, Seoul, Korea, in 1989 and 1991, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1997. Since 2000, he has been with the Department of Mechanical Engineering, Texas A&M University (TAMU), College Station, where currently he is Associate Professor and was the inaugural Holder of the Dietz Career Development Professorship II in 2007–2010. His current research interests include the analysis, design, and real-time control of mechatronic systems, networked control systems, and nanoscale engineering and technology. He is the holder of three U.S. patents on precision positioning systems. He is Fellow of ASME and Member of Pi Tau Sigma. Prof. Kim is Editor of *International Journal of Control, Automation, and Systems* and Associate Editor of *Asian Journal of Control*. He was Technical Editor of IEEE/ASME TRANSACTIONS ON MECHATRONICS and *ASME Journal of Dynamic Systems, Measurement and Control*.