

Article

Remote Control for Robot Walking via Gesture Detected by Inertial Sensor

Na Xiang¹, Fa-fa Wang¹, Bin-bin Wang¹, Sheng-lun Yi¹, Xue-bo Jin^{1,*}, Ting-li Su¹, Jian-lei Kong¹ and Yu-ting Bai²

¹ School of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100048, China; xiangna@st.btbu.edu.cn (N.X.); wangfafa@st.btbu.edu.cn (F.W.); wangbinbin@st.btbu.edu.cn (B.W.); yishenglun@st.btbu.edu.cn (S.Y.); sutingli@btbu.edu.cn (T.S.); kongjianlei@btbu.edu.cn (J.K.)

² School of Automation, Beijing Institute of Technology, Beijing 100081, China; byting@bit.edu.cn

* Correspondence: jinxuebo@btbu.edu.cn; Tel.: +86-136-9159-5989

Abstract: Wearable technology has been proposed as a potential tool to change the way of human life, such as the smart bracelet and the Google Glass. In the wearable technology, the inertial sensor has great significance in tracking the object movements. The paper focused on detecting the movements of user's finger based on the inertial sensor to give the control signals. Firstly, the attitude matrix, which represented the transformation relation of carrier coordinate system and the navigation coordinate system, was obtained. Secondly, the attitude matrix was expressed based on the quaternions. Thirdly, the finger gesture was processed by the attitude matrix to get the attitude angle. Finally, the robot was controlled by attitude angle to make the moving action. The experimental results showed the detection of the finger movement is effective.

Keywords: inertial sensor; finger gesture; NAO humanoid robot; quaternions; motion capture

1. Introduction

Nowadays, the wearable technology has drawn the attention of researchers and engineers to improve the HMI (Human Machine Interface), a useful product to provide convenience. Meanwhile, the inertial sensor with powerful data processing ability and a wearable advantage has been applied into lots of domains. In medical field, the researchers developed the automatic computer-aided balance assessment system. This method employed ambient assisted living architecture based on inertial sensors to examine the balance ability of the patients [1]. Similarly, the inertial sensor is also used in a wearable body fall monitoring device [2]. When the user's body posture changes greatly, the inertial sensor will automatically detect the change of angle to judge whether the user falls. Furthermore, the device will alarm by wireless communication device. In addition, a variability analysis of upper limb therapeutic movements using wearable inertial sensors is presented [3]. The objective quantification of human movements contributes to better understanding the patient condition, and might serve as a measure of efficacy of the rehabilitation treatment [4]. In sports, the estimation process of limb lengths is automated with the algorithm calculating curvature using the measurements from inertial sensors [5]. Furthermore, the inertial sensor can be also used to detect the human gait. It measured the acceleration information of human walking, calculating the pace through the software algorithm precisely [6]. In science and technology, gesture recognition is often explored for applications such as hand written character recognition with a handheld pen [7], or a handheld controller to operate a television [8]. The researchers examined gesture recognition algorithms that use an inertial sensor worn on the forearm to get the gestures. A set of six gestures were selected fit within the context of an active video game instead of using the mouse [9].

Compared with their devices, we designed a device which uses the inertial sensor to detect the change of the finger's gesture and send commands to the robot through wireless communication. Because the angle of finger is much smaller, we have to design the algorithm which had a higher precision and faster processing speed to realize the function of detecting the change of finger gesture.

One method describes the relative position between the navigation coordinate system and the carrier coordinate system is based on Euler angles [10][11][12], in which the attitude angle is calculated by solving the differential equations. However, the computer numerical integration will carry out a large number of trigonometric functions. Furthermore, it results in a 'singularity' problem and the inaccuracy of robot control. In this paper, we use quaternions to calculate the attitude matrix to get the attitude angle. The method just needs four differential equations to solve with less calculation. Moreover, it can avoid into the 'singularity' problem and improve the efficiency of the calculation. For the applications, the gesture of finger is captured to control the movement of NAO robot.

The content of this paper is divided into the following parts: Section 2 introduces how to detect finger gesture and process the data of finger gesture. Section 3 discusses the threshold selection and arranges experiments to demonstrate the performance of developed system. Section 4 gives the conclusion and discusses further research.

2. Inertial Navigation Coordinate System and Its Transformation

2.1. Inertial Navigation Coordinate System and Its Transformation

Carrier coordinate system is called b system ($O_{x_b y_b z_b}$), the origin is in the carrier's center of gravity, and is fixedly connected with the carrier. The axis x_b point to the front along the longitudinal axis, in addition, the axis y_b point to the right along the traverse axis. Furthermore, the axis z_b is perpendicular to the horizontal plane, which is composed of the axis x_b and y_b axis. Correspondingly, navigation coordinate system is called n system ($O_{x_n y_n z_n}$). The inertia components of the strap-down inertial navigation system (SINS) are fixed on the carrier, so the output value of the sensor is the output value of the carrier coordinate system.

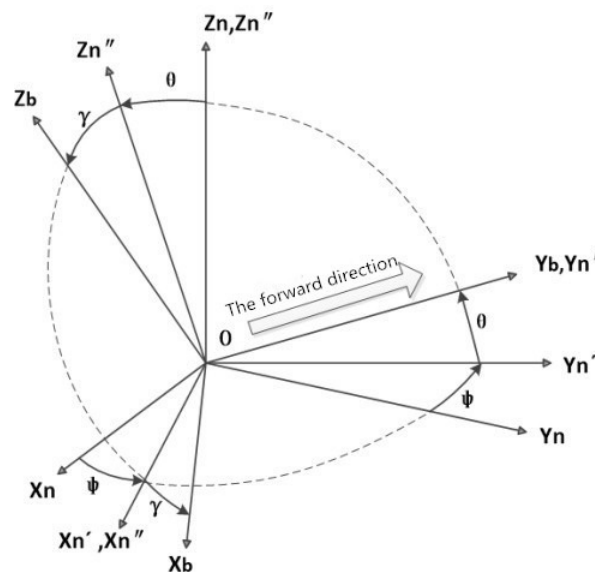


Figure 1. The transformation relation

Therefore, we must transform these measurements to another coordinate system for calculate the navigation parameters, that is, the navigation coordinate system.

The data from the carrier coordinate system to the navigation coordinate system must be transformed before the strap-down algorithm. As is shown in Figure 1, we can use three times of Euler rotation to transform the carrier coordinate system to the navigation coordinate system. It means that the navigation coordinate system needs to rotate ψ degree counterclockwise around the z_n axis. Secondly, it needs to rotate θ degree counterclockwise around the x_n' axis. Finally, we need to

make the navigation coordinate system rotate around y_n'' axis for about γ degree. The transition from navigation coordinate system to the carrier coordinate system is made in the following formula (1).

$$x_n y_n z_n \xrightarrow{\text{turn } \psi \text{ around } z_n} x_n' y_n' z_n' \xrightarrow{\text{turn } \theta \text{ around } x_n'} x_n'' y_n'' z_n'' \xrightarrow{\text{turn } \gamma \text{ around } y_n''} x_b y_b z_b \quad (1)$$

where ψ, θ, γ represent yaw, pitch and roll, respectively. The transformation relation between the b system and n system can be expressed by the following.

$$\begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} = C_n^b \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \quad (2)$$

Three times of Euler rotation can create three cosine matrices, the direction cosine matrix C_n^b can be obtained by multiplying the three matrices with the formula(3)[13].

$$\begin{aligned} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} &= \begin{bmatrix} \cos \gamma & 0 & -\sin \gamma \\ 0 & 1 & 0 \\ \sin \gamma & 0 & \cos \gamma \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \\ &= \begin{bmatrix} \cos \gamma \cos \psi - \sin \gamma \sin \theta \sin \psi & \cos \gamma \sin \psi + \sin \gamma \sin \theta \cos \psi & -\sin \gamma \cos \theta \\ -\cos \theta \sin \psi & \cos \theta \cos \psi & \sin \theta \\ \sin \gamma \cos \psi + \cos \gamma \sin \theta \sin \psi & \sin \gamma \sin \psi - \cos \gamma \sin \theta \cos \psi & \cos \gamma \cos \theta \end{bmatrix} \begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} \end{aligned} \quad (3)$$

The direction cosine matrix C_b^n is an orthogonal matrix, that is, $C_b^n = [C_n^b]^{-1} = [C_n^b]^T$, C_b^n can be obtained in the following.

$$C_b^n = \begin{bmatrix} \cos \gamma \cos \psi - \sin \gamma \sin \theta \sin \psi & -\cos \theta \sin \psi & \sin \gamma \cos \psi + \cos \gamma \sin \theta \sin \psi \\ \cos \gamma \sin \psi + \sin \gamma \sin \theta \cos \psi & \cos \theta \cos \psi & \sin \gamma \sin \psi - \cos \gamma \sin \theta \cos \psi \\ -\sin \gamma \cos \theta & \sin \theta & \cos \gamma \cos \theta \end{bmatrix}$$

To express simply, we rewrite C_b^n as $\begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}$, where

$$\begin{aligned} T_{11} &= \cos \gamma \cos \psi - \sin \gamma \sin \theta \sin \psi \\ T_{12} &= -\cos \theta \sin \psi \\ T_{13} &= \sin \gamma \cos \psi + \cos \gamma \sin \theta \sin \psi \\ T_{21} &= \cos \gamma \sin \psi + \sin \gamma \sin \theta \cos \psi \\ T_{22} &= \cos \theta \cos \psi \\ T_{23} &= \sin \gamma \sin \psi - \cos \gamma \sin \theta \cos \psi \\ T_{31} &= -\sin \gamma \cos \theta \\ T_{32} &= \sin \theta \\ T_{33} &= \cos \gamma \cos \theta \end{aligned} \quad (4)$$

2.2. Attitude Matrix Algorithm Based on Quaternions

In order to solve the real-time attitude angle, the current attitude matrix C_b^n must be determined. In addition, the quaternions can solve the problem of different spatial coordinate transformation. The method needs to solve the quaternion differential equation when calculating the attitude matrix which is described by the quaternion[14]. The following formula (5) is described

quaternion differential equation, where $q(t)$ represents quaternion, the symbol \circ indicates quaternion multiplication.

$$\dot{q}(t) = \frac{1}{2}q(t) \circ \omega_{nb}^b \quad (5)$$

The formulas (6) (7) are expressed as matrix form, that is, where ω_{nb}^b is a matrix expression of three-axis angular velocity [15].

$$\dot{q}(t) = \frac{1}{2}M'(\omega_{nb}^b)q(t) \quad (6)$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -w_{nbx}^n & -w_{nby}^n & -w_{nbz}^n \\ -w_{nbx}^n & 0 & -w_{nbz}^n & -w_{nby}^n \\ -w_{nby}^n & -w_{nbz}^n & 0 & -w_{nbx}^n \\ -w_{nbz}^n & -w_{nby}^n & -w_{nbx}^n & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \quad (7)$$

The orientation relationship between the carrier coordinate system and the navigation coordinate system can be expressed uniquely by quaternion. The following formula (8) shown conversion relationship:

$$\begin{bmatrix} x_n \\ y_n \\ z_n \end{bmatrix} = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix} \quad (8)$$

The posture matrix C_b^n expressed in the form of quaternion is given by the following formula (9):

$$C_b^n = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 - q_0q_3) & 2(q_1q_3 + q_0q_2) \\ 2(q_1q_2 + q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 - q_0q_1) \\ 2(q_1q_3 - q_0q_2) & 2(q_2q_3 + q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \quad (9)$$

The quaternion attitude matrix C_b^n in formula(9) is exactly equivalent to the direction cosine matrix C_b^n in formula(4). Here, the elements of the same position are equal, but the expression forms are different. The equivalent relationship is shown in the formula (10).

$$\left. \begin{aligned} q_0^2 + q_1^2 - q_2^2 - q_3^2 &= T_{11} \\ q_0^2 - q_1^2 + q_2^2 - q_3^2 &= T_{22} \\ q_0^2 - q_1^2 - q_2^2 + q_3^2 &= T_{33} \end{aligned} \right\} \quad (10)$$

where T_{11} , T_{22} , T_{33} is an element in the direction cosine matrix in formula(4), and q_0 , q_1 , q_2 , q_3 are the elements in the quaternion attitude matrix.

The elements of the cosine matrix can be calculated directly from the elements of the quaternion. Therefore, the attitude angle is solved by the method of inverse trigonometric function as shown in the formula (11).

$$\left. \begin{aligned} \theta &= \arcsin T_{32} \\ \gamma &= \arctan \frac{-T_{31}}{T_{32}} \\ \psi &= \arctan \frac{-T_{12}}{T_{22}} \end{aligned} \right\} \quad (11)$$

where ψ , θ , γ represent yaw, pitch and roll.

2.3. Process from Data of Finger Gesture to Movement of the robot

In this remote system, four control instructions correspond to the four types of finger gestures. The finger gestures include the finger lifts up, points to the front and parallels to the ground, turns right and turns left. The finger movements are respectively defined as *Finger_Up*, *Finger_Horizontal*

, *Finger_Right* and *Finger_Left*. Because these finger movements mainly change the values of Pitch and Yaw, only these two values are considered. When an operator makes the finger movements mentioned above, the gesture data are recorded by inertial sensor, as shown in Figure 2. The blue line and red line represent Pitch and Yaw. In order to accurately detect the finger gestures, we select the finger points to the front and parallels to the ground as the reference position. In addition, during the initialization, the initial angles of Pitch and Yaw are set as 0 degree. The finger movements can lead to the change of angle, and then the robot can be controlled. In order to detect the change of the angle accurately, the thresholds T_1 and T_2 are selected. The ideal discrimination rules are shown in the following formula (12).

$$\begin{cases} \text{Pitch} > T_1 \Rightarrow \text{Finger_Up} \\ \text{Pitch} = 0 \Rightarrow \text{Finger_Horizontal} \\ \text{Yaw} > T_2 \Rightarrow \text{Finger_Right} \\ \text{Yaw} < -T_2 \Rightarrow \text{Finger_Left} \end{cases} \quad (12)$$

where *Finger_Up* describes the finger lifts up, *Finger_Horizontal* describes the finger points to the front and parallels to the ground, *Finger_Right* describes the finger turns right, *Finger_Left* describes the finger turns left.

The relation of the gesture and the control of robot walking is as the following: When the user lifts his/her finger, we will find that the blue line goes up sharply and appears a peak, and then the robot is controlled to go forward. When the finger comes back to the reference position, the blue line will come back to the 0 degree and the robot will be controlled to stop directly. In addition, when the user's finger turns right, the red line will go up, and then the robot will be controlled to go right. Furthermore, when the user's finger turns left, the red line will decline quickly and emerge a valley. The robot will be controlled to turn left. The changes of the angle and the movements of the robot are shown in Figure 3.

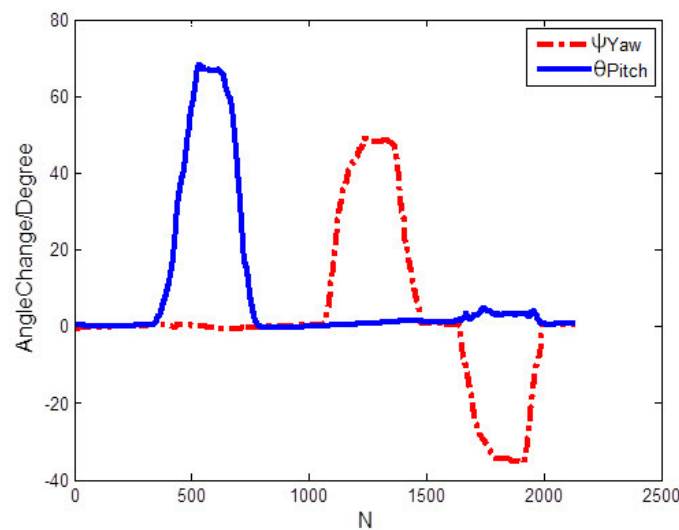


Figure 2. Measured data of all kinds of finger movements

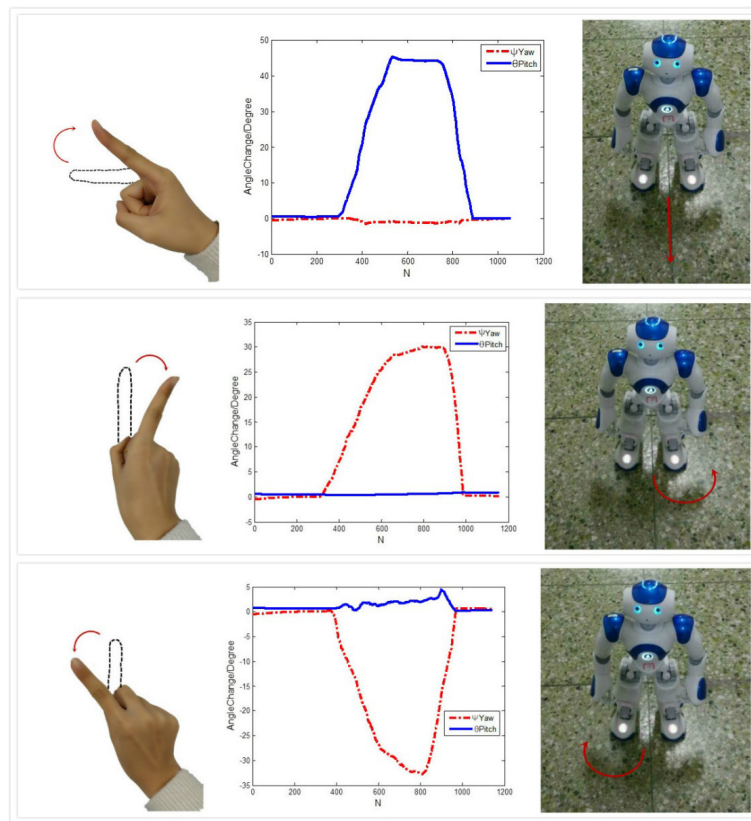


Figure 3. The changes of the angle and the movement of the robot.

3. Threshold Selection and Experiment

3.1. The Threshold Analysis

Due to the inertial sensor is set on the finger, the measurements will contain the disturbance and noise. Thus we should set the minimum value of angle to avoid the influence of noise. On the other hand, when the operator lifts his or her finger but without the attention to control the robot movement, the data will be measured by the inertial sensor, too. To avoid this situation, we set a maximum value to prevent the wrong movements. If the angle is large enough beyond the maximum value, the robot will also remain stationary.

In order to find out the reliable thresholds, we use the off-line experiments to analyze the measurements. During the experiment, the operator makes the corresponding finger gestures, as shown in Figure 4. These gestures will be transformed to corresponding instructions including *Finger_Up*, *Finger_Horizontal*, *Finger_Right* and *Finger_Left*. The recorded data of angle changes are shown in Figure 5 and Figure 6.

Figure 5 shows that the range of Pitch between 0 degree and 30 degrees may be caused by the slight jitter of the finger, which will give a wrong command. So we set the minimum angle T_1 of Pitch as 30 degrees, which indicated by the green line. And we set the maximum angle T_2 as 60 degrees, which indicated by the yellow line in the Figure 5. Therefore, the thresholds of 30 degrees to 60 degrees can be selected which can reflect the change of Pitch angle.



Figure 4. The finger fixed on the inertial sensor.

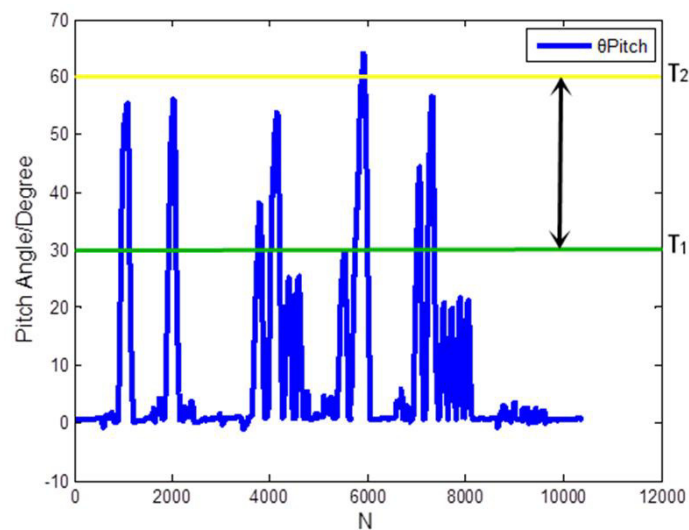


Figure 5. The range of Pitch angle.

When the finger turns right, the angle of Yaw will be a positive degree. On the contrary, when the finger turns left, the angle will be a negative degree. Similar to the Pitch, we select the threshold T_3 as ± 20 degrees and T_4 as ± 40 degrees to accurately reflect the change of Yaw. In the Figure 6 the green lines indicate T_3 and the yellow lines indicate T_4 . The rules to detect the movement of finger are as shown in the formulas (13) (14).

$$\begin{cases} T_1 < Pitch < T_2 \Rightarrow Finger_Up \\ Pitch = 0^\circ \Rightarrow Finger_Horizontal \end{cases} \quad (13)$$

$$\begin{cases} T_3 < Yaw < T_4 \Rightarrow Finger_Right \\ -T_4 < Yaw < -T_3 \Rightarrow Finger_Left \end{cases} \quad (14)$$

We note that Pitch belongs to $[T_1, T_2]$, it means that the finger is lifted up. Pitch is equal to 0 degree, it means that the finger is pointed to the front. Yaw belongs to $[T_3, T_4]$, it means that the finger is turned right. Yaw belongs to $[-T_4, -T_3]$, it means that the finger is turned left.

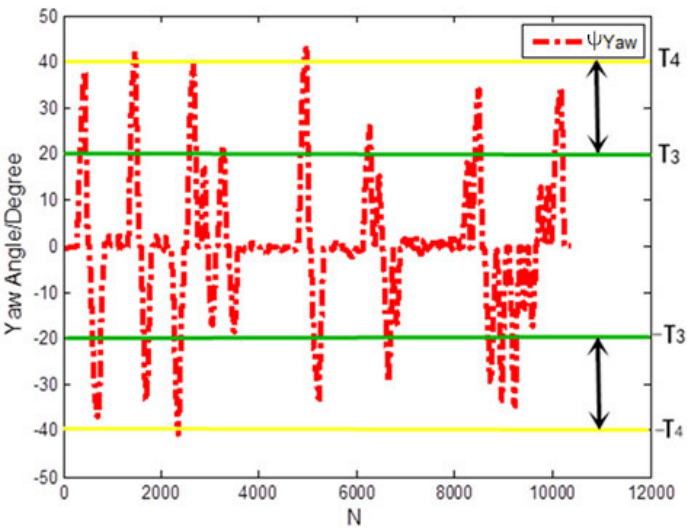


Figure 6. The range of Yaw angle.

3.2. Controller for NAO robot

A controller which runs on the robot is developed using the python language under the windows environment. It maintains two functions running on different threads: instruction receiving and behavior performing. Here, the robot acts as the server port, while the inertial sensor is the client port. The server receives the instruction from the client with the form of input stream. At the same time, the controller gives four types of character instructions: "G", "S", "L" and "R" whose definitions are listed in Table 1. During the entire control process, the robot needs to receive the command from the bound port continuously, so that it is able in time to obtain the instruction from the client.

Table 1. Definition of commands.

Command	Definition
G	controlling the robot to go forward
S	controlling the robot to stop
L	controlling the robot to turn left
R	controlling the robot to turn right

3.3. Robot Movement

In order to verify the accuracy of the design of the system, we use the finger position of the navigation robot from the initial position to reach the destination. First of all, we establish the relationship between the inertial sensor and the robot. Then, the robot will be controlled to go forward, stop, turn right and turn left by the inertial sensor which is fixed on the operator’s finger. The angle of finger measured by the inertial sensor as shown in Figure 7 and the experimental scene of the robot is shown in Figure 8.

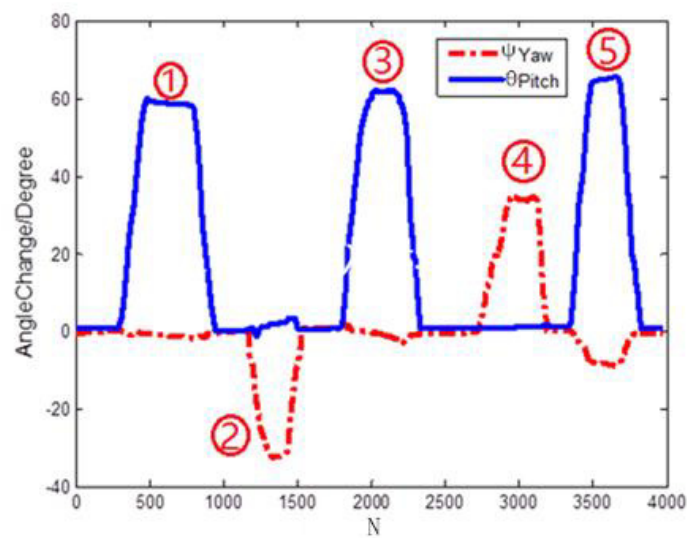


Figure 7. The angle of finger measured by the inertial sensor.

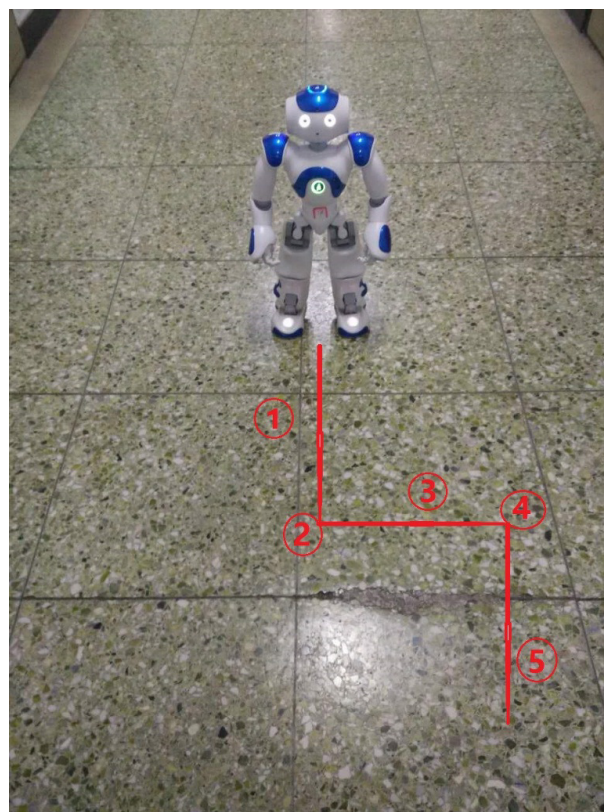


Figure 8. Experimental scene.

4. Conclusion

The integration between wearable technology and robot control based on the inertial sensor is considered in this paper. A method was proposed for remote control of robot via gesture detection based on inertial sensor. The system uses the inertial sensor to detect the moving direction of the finger and sends corresponding commands to the robot through wireless communications. In order

to improve the accuracy, quaternion is used in our method. Experiment verified the effectiveness and reliability of the proposed method.

There are some significant advantages of this method as follows:

- 1) The system based on inertial sensor is more portable due to the wearable feature, and the control command switching is more natural and smoother.
- 2) This finger control system can use the finger gestures to control the robot.
- 3) It shows a better precision compared with other approaches and makes the operation more convenient.

Acknowledgments: This work is partially supported by NSFC under Grant No. 61273002, 61673002 and the Key Science and Technology Project of Beijing Municipal Education Commission of China No. KZ201510011012.

References

1. Badura, P.; Pietka, E. Inertial Sensor Location Analysis in Automatic Balance Assessment. In *Proceedings of the 2015 IEEE 22nd International Conference Mixed Design of Integrated Circuits and Systems*, Torun, Poland, 25-27 June 2015; pp.48-52.
2. Shastry, M.; Asgari, M.; Wan E.A, et al. Context-Aware Fall Detection Using Inertial Sensors and Time-of-Flight Transceivers. In *Proceedings of the IEEE 38th Annual International Conference of the Engineering in Medicine and Biology Society*, Orlando, Florida, USA, 16-20 August 2016; pp.570-573
3. López-Nava, I.H.; Arnrich, B.; Muñoz-Melndez, A. et al. Variability Analysis of Therapeutic Movements using Wearable Inertial Sensors. *Journal of Medical Systems*. **2017**, *41*, 1-19.
4. Miloš, D.; Kostić, Maša, D. Popović. The modified drawing test for assessment of arm movement quality. *Automatic Control*. **2013**, *21*, 49-53
5. Karunarathne, M. S.; Li S.Y.; Ekanayake, S.W. et al. A Machine-Driven Process for Human Limb Length Estimation Using Inertial Sensors. In *Proceedings of the IEEE 10th International Conference on Industrial and Information Systems*, Peradeniya, Sri Lanka, 17-20 Dec 2015; pp.429-433
6. Xuan, Y. D.; Sun, Y.F.; Huang Z.B. et al. Step Cycle Detection of Human Gait Based on Inertial Sensor Signal. In *Advances in Wireless Sensor Networks*, Sun L.M., Ed.; Springer Heidelberg, 2014; Volume 501; pp.97-104.
7. Zhou, S.L.; Dong, Z.X.; Li, W.J. Hand-Written Character Recognition Using MEMS Motion Sensing Technology. In *Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. Xian, China, 2-5 July 2008; pp.1418-1423
8. Yang, J.; Choi, E.; Chang, W. et al. A Novel Hand Gesture Input Device Based on Inertial Sensing Technique. In *proceeding of the IEEE 30th Annual International Conference of Industrial Electronics Society*. Yokohama, Japan, 9-12 November 2015; pp.2786-2791
9. Arsenault, D.; Whitehead, A. D. Gesture Recognition Using Markov Systems and Wearable Wireless Inertial Sensors. *IEEE Transactions on Consumer Electronics*. **2015**, *61*, 429-437.
10. Diebel, J. Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. *Matrix*. **2006**, *58*.
11. Janota, A.; Šimák, V.; Nemec, D. et al. Improving the Precision and Speed of Euler Angles Computation from Low-Cost Rotation Sensor Data. *Sensors*. **2015**, *15*, 7016-7039.
12. Pittelkau, M. E. Rotation Vector in Attitude Estimation. *Guidance Control Dynamics*. **2003**, *26*, 855-860
13. Ran, C.; Cheng, X. A Direct and Non-Singular UKF Approach Using Euler Angle Kinematics for Integrated Navigation Systems. *Sensors*. **2016**, *16*, 1415-1438.
14. Goldman R. Understanding Quaternions. *Graphical Models*, **2011**, *73*, 21-49.
15. Yuan. X.; Yu. S. Yu.; Zhang, S. et al. Quaternion-based unscented kalman filter for accurate indoor heading estimation using wearable multi-sensor system. *Sensors*, **2015**, *15*, 10872-10890.



© 2016 by the authors; licensee Preprints, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).