Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.yxpublications.com/ijamce/index.html

Online Motion Pattern Recognition of Finger Gesture by Inertial Sensor

Xuebo Jin[a,*], Na Xiang[a], Tingli Su

[a] *School of Computer and Information Engineering, Beijing Technology and Business University, Beijing, 100048, China*

ARTICLE INFO

Article history:
Received 24 June 2017
Accepted 11 September 2017
Available online 25 December 2017

Keywords:
Inertial sensor
Finger gesture
Quaternions
Motion pattern
Online recognition

ABSTRACT

The inertial sensor can capture slight motion and has been used in many systems, such as robot command systems, hand-written character recognition, etc. Compared to classical pattern recognition systems, motion pattern recognition systems require online processes, which results in difficulties in the signal processing. This paper focused on detecting the movements of a user's finger based on the inertial sensor. Firstly, the finger gesture was captured by the inertial sensor, and then the motion pattern was identified based on filtering and attitude calculations. Finally, to verify the effectiveness of the method, the different motion pattern was sent to the robot as the movement command and the robot was controlled to move based on the attitude angle of the inertial sensor. Experimental results showed that the detection of finger movement was effective and accurate.

1. Introduction

Motion recognition has attracted a great deal of research interests in recent years because of its wide application. As the common motion recognition system, gesture recognition has been used in human-computer interaction, machine vision, robot control, etc.

In practice, there are 2 main methods of gesture recognition, and one of them is vision-based gesture recognition (e.g., Rautaray S S, Agrawal A. 2015; Biswas K K, Basu SK. 2011; Aggarwal J K, Xia L. 2014.; Chen C, Jafari R, Kehtarnavaz N. 2016). However, this method is invalid while the line of sight is obstructed. Furthermore, there were limitations associated with the utilization of RGB(color) cameras(e.g., Badura, P. Pietka, E. 2015), which required having a considerable amount of hardware resources in order to run computationally intensive image processing and computer vision algorithm (e.g., Chen C, Jafari R, Kehtarnavaz N. 2016).

The other main method of gesture recognition is based on inertial sensor identification technology, which is not affected by the environment or light, the application including automatic assessment of patient balance (e,g., Badura, P.; Pietka, E. 2015), the human limb length estimation (e.g., Karunarathne, M. S.; Li S.Y.; Ekanayake, S.W. et al. 2015), gait detection (e.g., Xuan, Y. D.; Sun, Y.F.; Huang Z.B. et al. 2014), hand-written character recognition (e.g., Zhou, S.L.; Dong, Z.X.; Li, W.J.2008), or using a handheld controller to operate a television (e.g., Yang, J.; Choi, E.; Chang, W. et al. 2015), etc. According to different applications, there are two kinds of the finger gesture to be recognized by inertial sensor: series gesture (e.g., Zhou, S.L.; Dong, Z.X.; Li, W.J.2008; Arsenault, D.; Whitehead, A. D. 2015; Xu C, Pathak PH, Mohapatra P. 2015); and single gesture(e.g.,Badura, P.,Pietka, E. 2015.; Yang, J.; Choi, E.; Chang, W. et al. 2015;Xu C, Pathak PH, Mohapatra P. 2015.)

As to the first one, i.e., series gesture recognition, the so called the Hidden Markov Model (HMM) is the popular used method. The HMM is a doubly stochastic state machine that has a Markov distribution associated with the transitions across various states, and a probability density function that models the output for every state. Based on HMM a set of 6 gestures were selected to fit within the context of an active game instead of using a mouse(e.g., Arsenault, D.; Whitehead, A. D. 2015). And as noted by Schlomer T et al.(e.g., Yin J, Chen H, Yang M, et al.2015), the HMM was used to process the data of accelerometer of a Wii-controller to recognize the user's gestures. The method is also applied in (e.g., Yuan. X.; Yu. S. Yu.; Zhang, S. et al. 2015) and in the context of hand gesture recognition, each state could represent a set of possible hand positions.

On the other hand, it seems that few method is applied for accurate recognition of the single gesture, for example, Pathak and Mohapatra proposed the motion energy by which when the user makes an action, the energy produced by the finger, hand, and arm is identified. Through calculating motion energy measured by the smartwatch, the arm, hand or finger gestures is identified(e.g., Morteza Khodabin, Majid Rostami. 2015.).

In this paper, we used the inertial sensor to detect the motion pattern of finger gestures and send commands to the robot by wireless communication. However, because the angle of the finger was much smaller compared with in other applications (e.g., Badura,

* Corresponding author.
E-mail addresses: jinxuebo@btbu.edu.cn (Jin Xue-bo)

P.; Pietka, E. 2015; Yang, J.; Choi, E.; Chang, W. et al. 2015), we had to design an algorithm with a higher precision and faster processing speed to realize the function of detecting changes in finger gestures. And unlike (e.g., Morteza Khodabin, Majid Rostami. 2015), the problem that we need to solve is identifying different gestures of the same part, i.e., the finger, therefore the motion energy is the same, so the method (e.g., Morteza Khodabin, Majid Rostami. 2015) of motion energy is not applicable here.

In this paper, we used the quaternion method to calculate the attitude matrix and based on the threshold, the attitude angle is used to accurately identify the movements of the fingers. This method required only 4 differential equations, and thus, solving required less calculation. Moreover, our method could avoid the 'singularity' problem and improve calculation efficiency.

The content of this paper is divided into the following parts: Section 2 introduces how to calculate the attitude matrix based on the angular velocity obtained by the inertial sensor, which is the basic algorithm used in this paper to obtain accurate gesture recognition. Section 3 outlines the system structure, including the inertial sensor, computer, and robot, and also gives the definition of gestures and discusses how to recognize gestures by the inertial sensor, and then defines the threshold selection. Section 4 presents our experiments to demonstrate the performance of the developed system and show the movement of the robot. Finally, Section 5 presents conclusions and discusses further research.

2. Calculation of Attitude Angle

In order to obtain the accurate attitude angle, an attitude matrix must be processed. One method of calculating attitude angle is to use the Euler algorithm, by which each differential equation contains a large number of trigonometric functions. However, the equation has a "singularity" problem, and the computing speed is slow (e.g., Diebel J. 2006). Another method is a direction cosine method, in which 9 differential equations must be calculated, and leads to large amount of calculations and poor real-time performance (e.g., Yin J, Chen H, Yang M, et al.2015). Finally, the quaternion method is represented by just 4 scalars, and the calculation of trigonometric functions is effectively avoided. In addition, it has the advantage of no singularity when the angle is approaching 90 degrees, and has a higher computation efficiency than the Euler angle and direction cosine matrix (e.g., Yuan. X.; Yu. S. Yu.; Zhang, S. et al. 2015). Thus, in this paper, we used the quaternion method.

The method of calculating the attitude matrix by the quaternion method is as follows:

1) Initialization of quaternions by the initial attitude.

$$q=\begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T, q_0{}^2+q_1{}^2+q_2{}^2+q_3{}^2=1 \quad (1)$$

2) The angular velocity values of each axis $\omega_x, \omega_y, \omega_z$ are measured by the inertial sensor. Construct Eq. (2) to get the updated value of quaternions.

$$\dot{q}(\mathrm{t})=\frac{1}{2}M(\omega)q(\mathrm{t}) \quad (2)$$

$$M(\omega)=\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & -\omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \quad (3)$$

3) The fourth-order Runge-Kutta method is used to solve the differential Eq.(2). As the following (e.g., Morteza Khodabin, Majid Rostami. 2015):

$$\begin{cases} K_1=\Gamma(t)q(t) \\ K_2=\Gamma(t+\frac{h}{2})(q(t)+\frac{h}{2}K_1) \\ K_3=\Gamma(t+\frac{h}{2})(q(t)+\frac{h}{2}K_2) \\ K_4=\Gamma(t+\frac{h}{2})(q(t)+hK_3) \\ q(t+h)=q(t)+\frac{h}{6}(K_1+2K_2+3K_3+K_4) \end{cases} \quad (4)$$

where K represents the slope, t is the present moment, h represents the update step, and $M(w)$ is the angular velocity matrix expression of the 3 axes.

4) After using the Runge-Kutta method to solve the differential equation and obtain $q(t)$, attitude matrix C can be expressed in the form of quaternions, as shown in Eq. (5).

$$C=\begin{bmatrix} q_0^2+q_1^2-q_2^2-q_3^2 & 2(q_1q_2-q_0q_3) & 2(q_1q_3+q_0q_2) \\ 2(q_1q_2+q_0q_3) & q_0^2-q_1^2+q_2^2-q_3^2 & 2(q_2q_3-q_0q_1) \\ 2(q_1q_3-q_0q_2) & 2(q_2q_3+q_0q_1) & q_0^2-q_1^2-q_2^2+q_3^2 \end{bmatrix} \quad (5)$$

To express this simply, we rewrite attitude matrix C as in Eq. (6).

$$C=\begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (6)$$

Then, the attitude angle is calculated by the method of inverse trigonometric functions, as shown in Eq. (7).

$$\left.\begin{aligned} \theta &= \arcsin(-C_{31}) \\ \gamma &= \arctan\frac{C_{32}}{C_{33}} \\ \psi &= \arctan\frac{C_{21}}{C_{11}} \end{aligned}\right\} \quad (7)$$

where ψ, θ, γ represent yaw, pitch, and roll, respectively. Pitch is the angle of rotation around the X axis (Fig. 1(a)), roll is the angle of rotation around the Y axis (Fig. 1(b)), and yaw is the angle of rotation around the Z axis (Fig. 1(c))(Bortolami S B, Pierobon A, Dizio P, et al. 2006). A flow diagram of the algorithm is shown in Fig. 2.

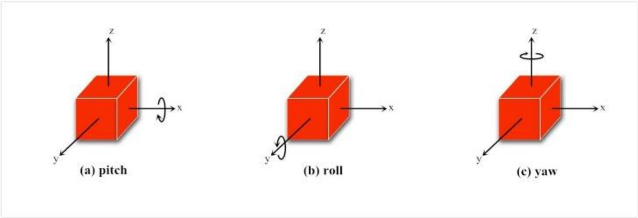


Fig. 1. Sketch map of pitch, yaw, and roll.

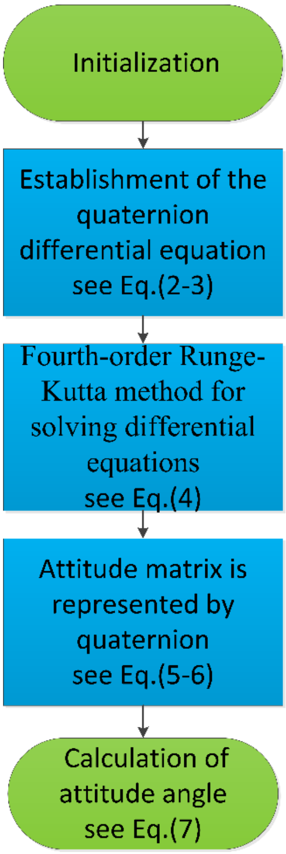


Fig. 2. Flow diagram of quaternion method.

3. Recognition of Finger Gesture by Inertial Sensor

3.1 Outline of System

Our remote-control system included an inertial sensor, the computer, and a Nao humanoid robot (Fig.3). The inertial sensor was produced by Beijing iGyro Technology Co., Ltd, which we applied to get the measured angular velocity used in Eq. (3). During the experiment, the inertial sensor was fixed on the user’s finger to detect the gesture.

The Nao robot was produced by Aldebaran Robotics. This company opened Nao technology to all higher-education projects, and the hardware design used the latest manufacturing technology to ensure smooth operation. Besides, the Nao was equipped with a variety of sensors. As a classical robot platform, the Nao humanoid robot can keep itself stable when walking, and each action has a corresponding program code. When the user wants the robot to move, it is necessary to send the corresponding program code to the robot through the computer.

In the gesture recognition system, finger movement was recognized by the measurement signal of the inertial sensor and sent to control the robot. The relation of the gesture and the robot’s walking command are shown in Fig. 4 and are described as follows:

(1) When the user's finger is lifted from the horizontal state, the robot is controlled to move forward;

(2) When the user's finger is pointed down, the robot is controlled to stop;

(3) When the user's finger turns right, the robot is controlled to turn right;

(4) When the user's finger turns left, the robot is controlled to turn left.

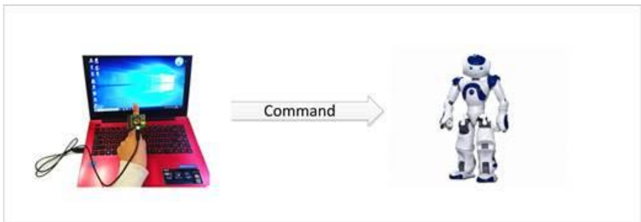


Fig. 3. The system with a Nao humanoid robot, an inertial sensor, and the computer.

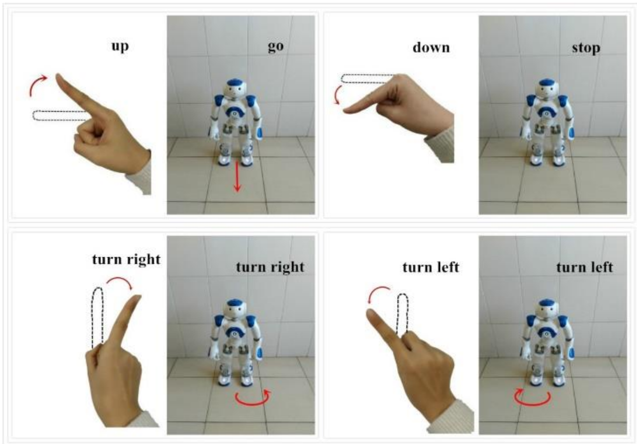


Fig. 4. Relation of gesture and robot movement

3.2 Gesture Recognition

To get the gesture pattern, we considered the measurement signal from the inertial sensors firstly. As we can see from Fig. 5-8, when the finger moves, 3 signals, called *X,Y,Z* axis were obtained. For example, when the finger was lifted, the angular velocity of *X* axis changed, as shown in Fig. 5, and the angular velocity was positive. When the finger returned to its original position, the angular velocity was negative. This process reflects a peak and a valley in the *X* axis, where the peak is in front of the valley. Fig. 5 shows 3 signals, including the *X*, *Y*, and *Z* axes, obtained when the users lifted their finger 3 times. We use the dashed box with the label ① ② ③ to indicate these 3 times shown in Fig. 5.

Similar to in Fig. 6, when the finger was turned below the horizontal position, the angular velocity of the *X* axis also changed. However, here, the angular velocity was firstly negative. When the finger returned to its original position, the angular velocity was turned positive. This process reflects a valley and a peak in the *X* axis, where the valley is in front of the peak.

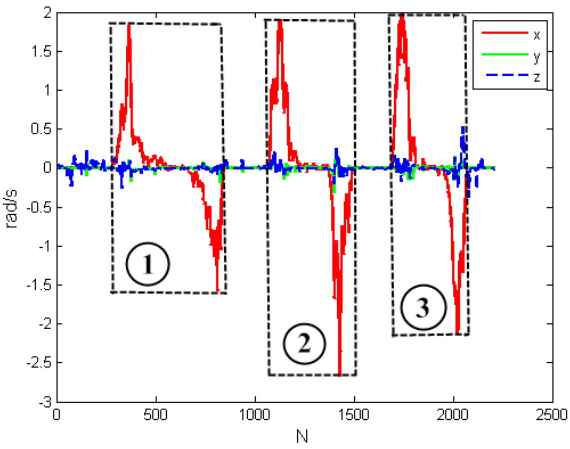


Fig. 5. Angular velocity signals obtained by inertial sensor when the finger is lifted 3 times

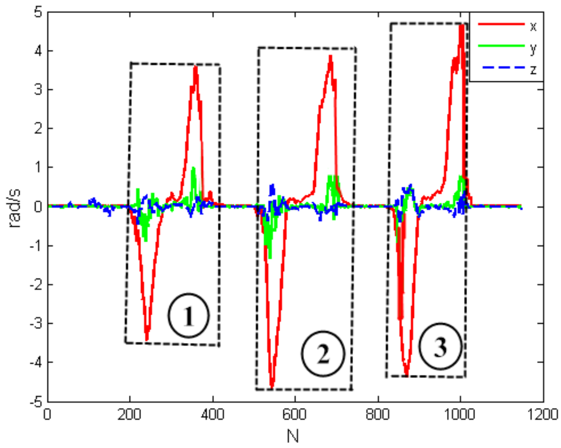


Fig. 6. Angular velocity signals obtained by inertial sensor when the finger is down 3 times

When the user's finger turned right, the angular velocity of the Z axis changed, as shown in Fig. 7, and the angular velocity was positive. When the finger returned to its original position, the angular velocity was negative. This process reflects a peak and a valley in the Z axis, where the peak is in front of the valley.

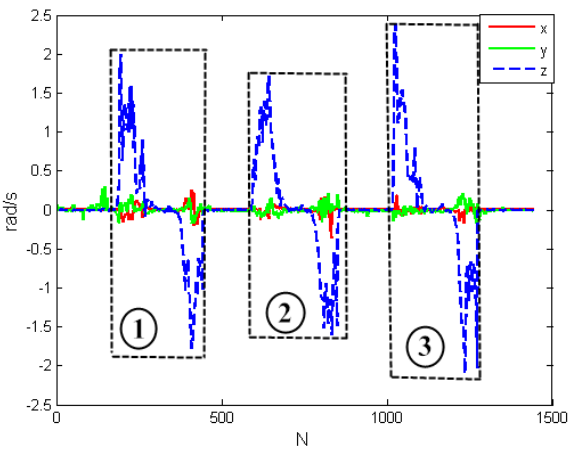


Fig. 7. Angular velocity signals obtained by inertial sensor when the finger turns right 3 times.

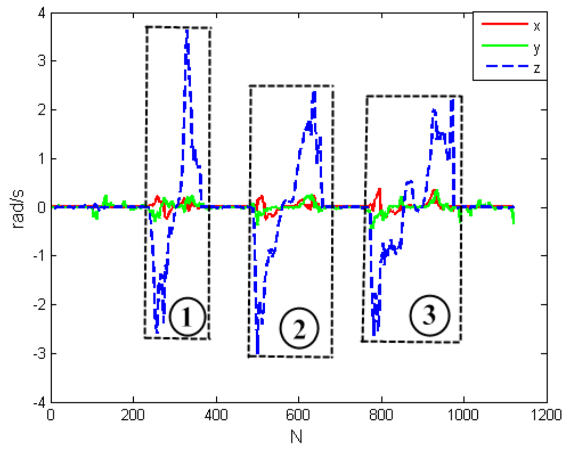
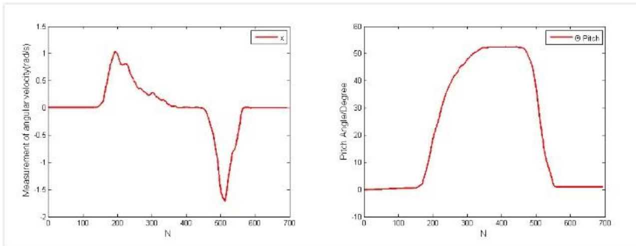
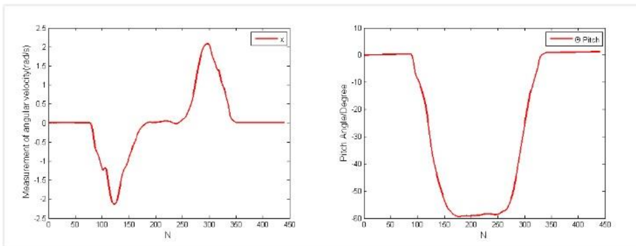


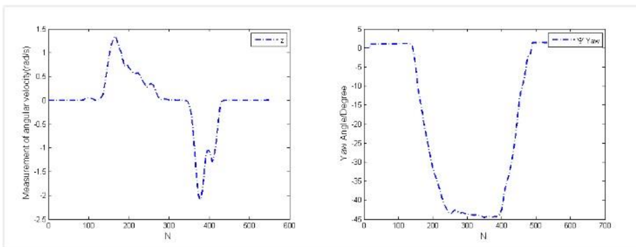
Fig. 8. Angular velocity signals obtained by inertial sensor when the finger turns left 3 times.



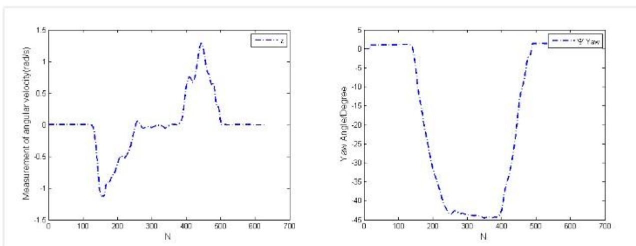
(a) Measurement of angular velocity and attitude angle when the finger is lifted



(b) Measurement of angular velocity and attitude angle when the finger is turned down



(c) Angular velocity signals obtained by inertial sensor when the finger turns left 3 times.



(d)Angular velocity signals obtained by inertial sensor when the finger turns left 3 times.

Fig. 9. Relation of gestures and attitude angles.

When the user's finger turned left, the angular velocity of the *Z* axis changed, as shown in Fig. 8, and the angular velocity was negative, and when the finger returned to its original position, the angular velocity was positive. This process reflects a peak following a valley in the Z axis.

From Fig. 5-8, we can see that when the finger makes some gestures, different signals are obtained, which makes it possible to obtain gesture patterns based on these signals. However, we did not obtain the pattern of the gestures based on the angular velocity signals directly in this paper. The reason for this is difficult to identify the series of peaks, such as the difference between Fig. 7 and Fig. 8; furthermore, the measurement signal would suffer noise, which would result in incorrect recognition. Therefore, we used the quaternion method to obtain the attitude angles.

The calculation method of the attitude angle discussed in Section 2 was used to obtain the angular velocity signal and attitude angle. Fig. 9(a) and 9(b) indicate the relationship between the angular velocity of the X axis and the pitch angle. In Fig. 9(a), pitch is positive, while in Fig. 9(b), pitch is negative. Fig. 9(c) and 9(d) indicate the relationship between the angular velocity of the Z axis and the yaw angle. In Fig. 9(c), yaw is positive, but is negative in Fig. 9(d). The sequence of the peak and valley here changes in regard to the different attitude angles, i.e., the pitch and yaw.

When a user makes finger movements (up, down, right, and left), gestures are obtained based on the attitude angle, as shown in Fig. 10. The red line and blue line represent pitch and yaw, respectively. As shown in the figure, when the user's finger lifted up, the angle of pitch was positive, and the red line had a positive peak (as labelled by ①.). When the user's finger is pointed down, the angle of pitch was negative, and the red line had a negative peak (as labelled by ②.). When the user's finger turned right, the angle of yaw was positive, and the blue line had a positive peak (as labelled by ③.). Lastly, when the user's finger turned left, the angle of yaw was negative, and the blue line had a negative peak (as labelled by ④.).

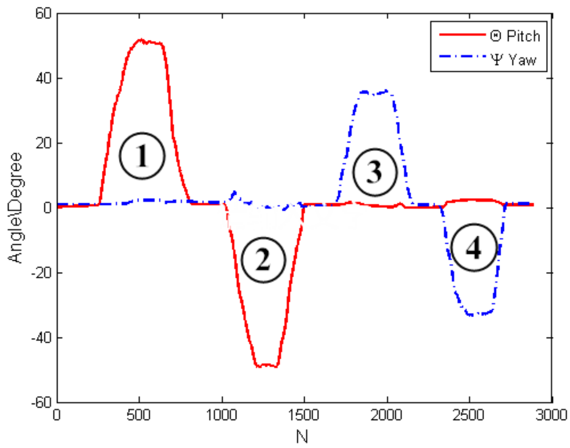


Fig. 10. Gesture and attitude angles with up, down, turn-right, and turn-left movements.

3.3 Threshold Analysis

Since the inertial sensor was fixed on the finger, slight movements of user's finger would result in gesture recognition inaccuracy. Thus, we set an angle change threshold to avoid the influence of noise. Regarding the recorded angle data shown in Fig. 11, we found that the range of pitch between -20 degrees and 20 degrees may have been caused by slight jittering of the finger. Thus, we set the minimum angle T_1 of pitch as ±20 degrees, as indicated by the green line in Fig. 11.

On the other hand, when the operator lifted their finger but without the intention to control the robot, a big attitude angle was obtained. To avoid this situation, the maximum angle T_2 was set as ±50 degrees, as indicated by the yellow line in Fig. 11. If the angle was beyond the maximum value, the robot would remain stationary. Therefore, the thresholds of ±20 degrees to ±50 degrees were selected, and reflect the angle of pitch.

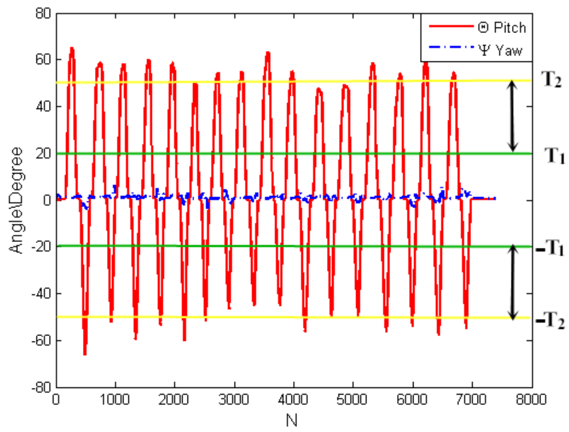


Fig. 11. Range of pitch angle.

When the finger turned right, the angle of yaw was positive. On the contrary, when the finger turned left, the angle was negative. Similar to the pitch, we selected thresholds T_3 as ±20 degrees and T_4 as ±40 degrees to accurately reflect the angle of yaw. In Fig. 12, the green lines indicate T_3 and the yellow lines indicate T_4 . The rules to detect finger movement are shown in Eq. (19) and (20).

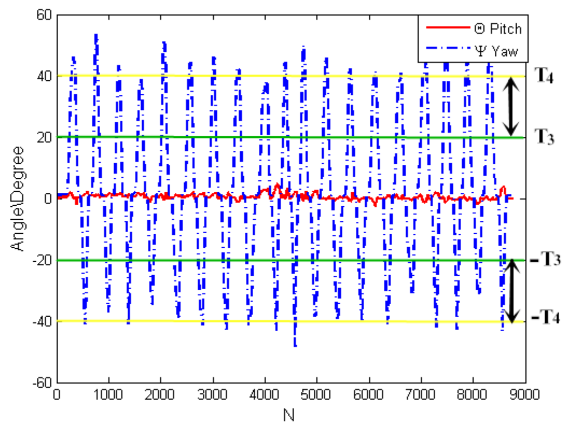


Fig. 12. Range of yaw angle.

$$\begin{cases} T_1 < Pitch < T_2 \Rightarrow Finger\_Up \\ -T_2 < Pitch < -T_1 \Rightarrow Finger\_Down \end{cases} \tag{19}$$

$$\begin{cases} T_3 < Yaw < T_4 \Rightarrow Finger\_Right \\ -T_4 < Yaw < -T_3 \Rightarrow Finger\_Left \end{cases} \tag{20}$$

Note that pitch belonging to [T_1,T_2] means that the finger was lifted up, pitch belonging to [T_1, T_2] means that the finger was down, yaw belonging to [T_3,T_4] means that the finger was turned right, and yaw belonging to [-T_4,-T_3] means that the finger was turned left.

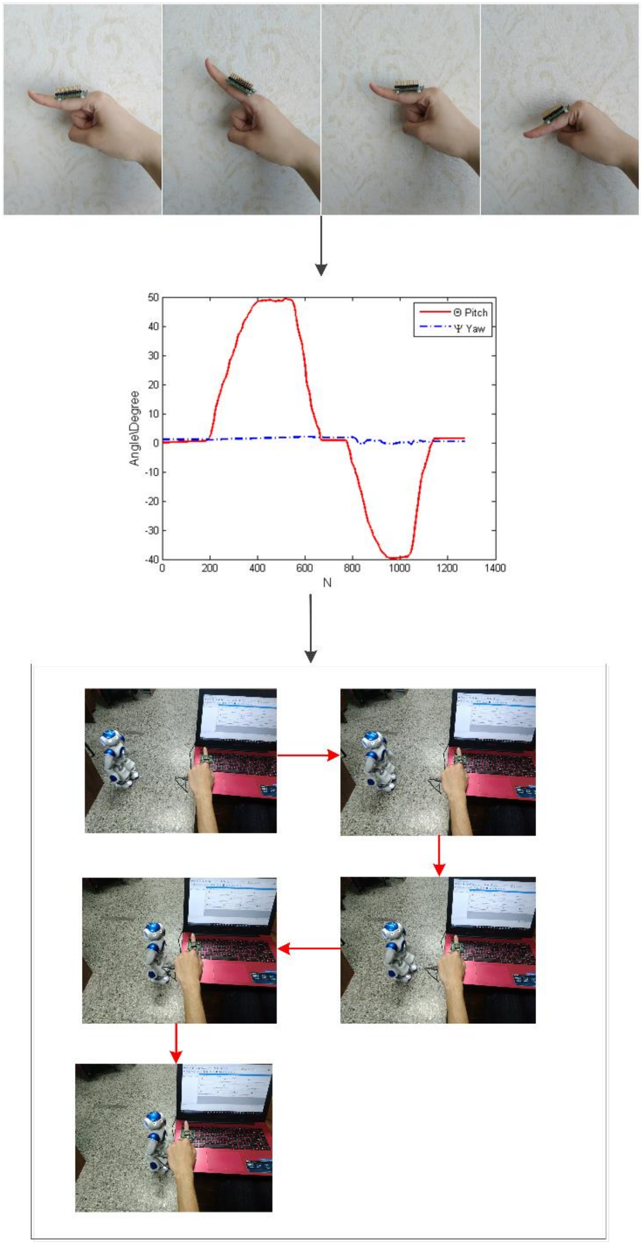
4. Control of the Nao Robot

The motion described in equations (19) and (20) was sent to the robot to command movement. As a classic robot platform, the Nao robot has a complete self-action module. With the movement command, the Nao can walk with balance, and therefore, we only considered whether the Nao could obtain accurate commands based on finger gestures. The computer obtained the signal from the inertial sensor, judged the finger gesture, and then gave the movement command to the Nao robot. The 4 commands described by the labels "G", "S'", "R", and "L", whose definitions are listed in Tab 1.

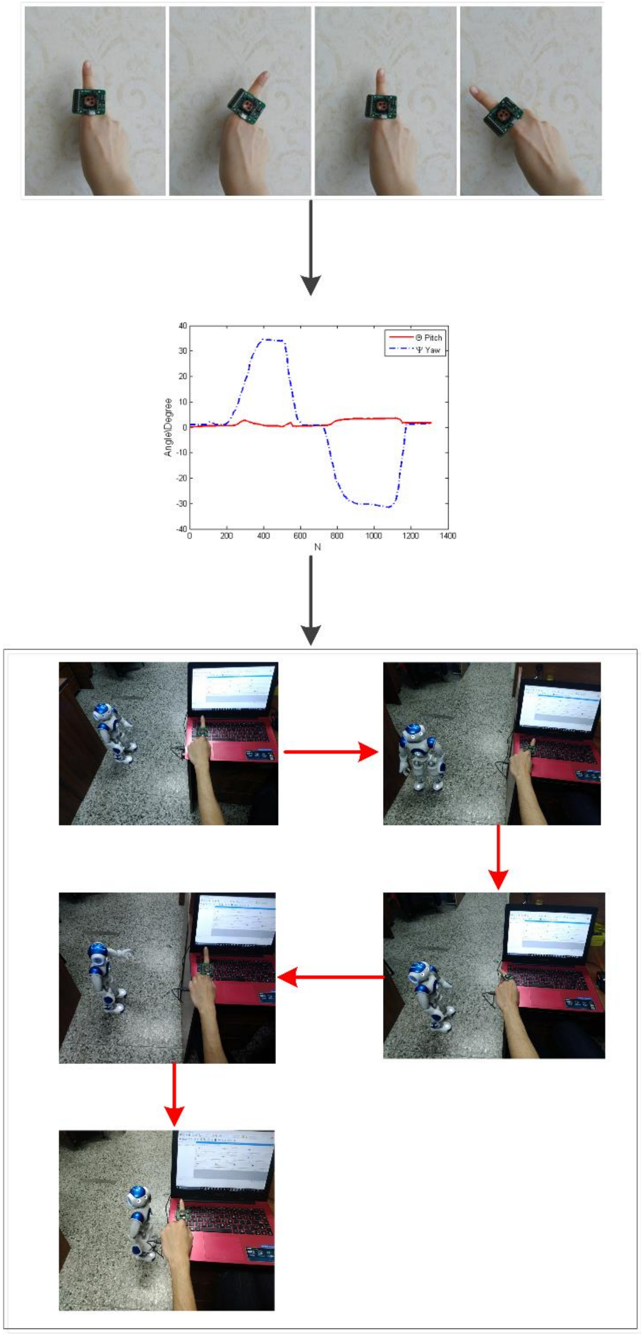
Tab 1. Definition of commands

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

As shown in Fig. 13, when the finger was lifted, pointed down, turned right, or turned left, the inertial sensor detected the change of the finger's angular velocity, which the quaternion method converted to the attitude angle. The robot then received a signal with this angle information, took the action corresponding to the command.



(a) Control the robot to go forward and stop



(d) Control the robot to turn right and left.

Fig. 13. Finger gestures and robot movements

As shown in Fig. 13(a), when the finger was lifted, the pitch had a positive angle and the robot was controlled to go forward. When the finger is pointed down, the pitch had a negative angle, the robot was controlled to stop. As shown in Fig. 13(b), when the finger was turned left, the yaw had a positive angle and the robot was controlled to turn right; meanwhile, when the finger was turned left, the yaw had a negative angle and the robot was controlled to turn left.

5. Conclusions

In this paper, we proposed a method for remote control of a robot via gesture detection based on an inertial sensor. The system used the inertial sensor to identify finger movement, and sent corresponding commands to the robot through wireless communications. Furthermore, to improve accuracy, we also employed the quaternion method. This method mainly used the fourth-order Runge-Kutta method to solve differential equation described by the quaternion, thus updating the quaternion to update the attitude angle. The result shows the robot can be controlled by the finger gestures based on the inertial sensor.

Acknowledgements

This work is partially supported by NSFC under Grant No. 61273002, 61673002, Beijing Natural Science Foundation No. 9162002 and the Key Science and Technology Project of Beijing Municipal Education Commission of China No. KZ201510011012.

References

Rautaray S S, Agrawal A. 2015.Vision Based Hand Gesture Recognition for Human Computer Interaction: A Survey. Kluwer Academic Publishers.

Biswas K K, Basu SK. 2011. Gesture Recognition Using Microsoft Kinect®. Automation, Robotics and Applications (ICARA), 2011 5th International Conference on IEEE, 100–103.

Aggarwal J K, Xia L. 2014. Human activity recognition from 3D data: A review. Pattern Recognition Letters. 48(1),70-80.

Chen C, Jafari R, Kehtarnavaz N. 2016. A Survey of Depth and Inertial Sensor Fusion for Human Action Recognition. Multimedia Tools and Applications, 1-21.

Badura, P.; Pietka, E. 2015. 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 48-52.

Karunarathne, M. S.; Li S.Y.; Ekanayake, S.W. et al. 2015. A Machine-Driven Process for Human Limb Length Estimation Using Inertial Sensors. In Proceedings of the 2015 IEEE 10th International Conference on Industrial and Information Systems. Peradeniya, Sri Lanka, 17-20, 429-433

Xuan, Y. D.; Sun, Y.F.; Huang Z.B. et al. 2014. Step Cycle Detection of Human Gait Based on Inertial Sensor Signal. In Advances in Wireless Sensor Networks, Sun L.M., Ed.; Springer Berlin Heidelberg, 501, 97–104.

Zhou, S.L.; Dong, Z.X.; Li, W.J.2008. Hand-Written Character Recognition Using MEMS Motion Sensing Technology. In Proceedings of the 2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Xian, China, 2-5,1418-1423

Yang, J.; Choi, E.; Chang, W. et al. 2015. A Novel Hand Gesture Input Device Based on Inertial Sensing Technique. In proceeding of the 2015 IEEE 30th Annual International Conference of Industrial Electronics Society. Yokohama, Japan, 9-12,2786-2791

Arsenault, D.; Whitehead, A. D. 2015. Gesture Recognition Using Markov Systems and Wearable Wireless Inertial Sensors. IEEE Transactions on Consumer Electronics. 61,429-437.

Schlomer T, Poppinga B, Henze N, Boll S. 2008.Gesture Recognition with a Wii Controller. Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction New York, NY, USA: ACM; 2008. p. 11–14.

Xu C, Pathak PH, Mohapatra P. 2015.Finger-writing with Smartwatch: A Case for Finger and Hand Gesture Recognition Using Smartwatch. Proceedings of the 16th International Workshop on Mobile Computing Systems and Applications New York, NY, USA: ACM. 9–14.

Wang SB, Quattoni A, Morency LP, Demirdjian D, Darrell T.2006. Hidden Conditional Random Fields for Gesture Recognition. 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

Diebel J. 2006.Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors. Matrix. 58.

Yin J, Chen H, Yang M, et al.2015. Analysis and Comparison of Direction Cosine Matrix and Quaternion Methods for Strapdown Inertial Navigation Attitude Algorithm[J]. Journal of Sichuan Ordnance, 9

Yuan. X.; Yu. S. Yu.; Zhang, S. et al. 2015, Quaternion-Based Unscented Kalman Filter for Accurate Indoor Heading Estimation Using Wearable Multi-Sensor System. Sensors. 15, 10872-10890.

Morteza Khodabin, Majid Rostami. 2015. Mean Square Numerical Solution of Stochastic Differential Equations by Fourth Order Runge-Kutta Method and Its Application in the Electric Circuits With Noise. Advances in Difference Equations, 1:1-19.

Bortolami S B, Pierobon A, Dizio P, et al. 2006.Localization of the Subjective Vertical During Roll, Pitch, and Recumbent Yaw Body Tilt. Experimental Brain Research.173(3):364-373.

Dr. Xue-bo Jin was born in Liaoning in 1972, China. She received the B.E. degree in industrial electrical and automation and the Master degree in industrial automation from Jilin University, Jilin, China, in 1994 and 1997, and the Ph.D. degree in control theory and control engineering from Zhejiang University, Zhejiang, China, in 2004. From 1997 to 2012 she was with College of Informatics and Electronics, Zhejiang Sci-Tech University. Since 2012 she has been with College of Computer and Information Engineering, Beijing Technology and Business University as a Professor. Her research interests include multisensor fusion, statistical signal processing, video/image processing, robust filtering, Bayesian theory, graph theory and Time series analysis, artificial intelligence, financial automation, target tracking and dynamic analysis. In particular, her present major interest is multi sensor fusion, Bayesian estimation and big data tendency analysis.

Na Xiang is currently a second-year graduate student in control engineering. In the past two years, she has studied several specialized courses, including system identification, optimal control and multisource information fusion. Her research interests include robotics, indoor positioning and navigation.

Ting-li Su received her Ph.D. degree from Beijing Institute of Technology(BIT), China, in 2013. Before that, she spent two and a half years as a visiting student in University of Bristol, UK, for her doctoral research work. Now she became a lecturer in Beijing Technology and Business University. Her research interests include state estimation, information fusion, signal processing and so on.