Control System for Wheeled Humanoid Following Robot

Baiyong Ding1, Shuaifei Ji1, Runping Han1,* and Lihong Chen2

1Chinese Fashion Science & Technology Research Institute, Beijing Institute of Fashion Technology, Beijing, China
2Fashion Accessory Art and Engineering College, Beijing Institute of Fashion Technology, Beijing, China

*Corresponding author

Abstract. Aiming at application and development prospects of following robot, a control system for humanoid 18-DOF robot supported by four wheels is designed. The system adopts the double-controller structure composed of master controller Raspberry Pi and slave controller STM32 for servo control of robot based on multi-sensor information fusion. In this control system, USB camera and two-axis pan-tilt are used for capturing images with wide viewing angles. Multi-scale template matching and Median-Flow algorithm are used for target recognition. Kalman Filter is used to fuse acceleration and angular velocity given by attitude sensor to obtain the reliable upright angle. PID algorithms are applied to the control system to realize robot’s upright angle control, speed control and turning angle control. Robot’s action adjustment is achieved by controlling its joint steering gears. The experimental results show that this low-cost wheeled humanoid robot has good self-balancing stability, anti-interference and following stability.

1. Introduction

With the progress and development of science and technology, highly integrated and intelligent robots represented by mobile service robots have shown their rapid development and huge application potential. As a kind of service robot, following robot can track target autonomously, which has a large number of applications in the fields of special personnel’s assistance, logistics and transportation, military and aerospace and so on[1]. Based on visual target recognition, robot can accurately identify targets and adapt to complex environment. However, the cutting-edge recognition algorithm usually needs a high-performance super-computing platform that has higher cost[2]. In addition, the environment that robot operate in the following process is complicated. In the case of complex environment such as slope and collision, it is necessary for robot to effectively control the balance of itself. In order to solve the above problems, a low-cost and highly adaptable wheeled humanoid robot control system is proposed. This system uses a simple image processing algorithm to achieve high-accuracy target recognition, and introduces the robot’s upright angle control and gravity center adjustment to make robot adapt to special situations such as slope and collision. Under the control of this system, robot can identify and track target quickly and smoothly.

2. Scheme of the Control System

The control system is used for the control of wheeled humanoid robot, which is composed of hardware system and software system. The hardware system design includes selecting appropriate controllers and sensors and designing peripheral circuits. The software system design involves the program design of algorithms needed to control robot by using Python and C languages.
The robot control system adopts double-controller\cite{3}, Raspberry Pi 4B is selected as master controller and STM32f103 as slave controller. According to the following strategy for robot, the master controller realizes acquisition and processing of image and controls joint steering gears of robot and outputs expected speed and expected turning angle of robot to slave controller. The slave controller obtains upright angle of robot, distance between robot and target, rotation speed of left and right wheels to realize upright angle control, speed control and turning angle control of robot. The structure diagram of the control system is shown in Fig.1.

![Figure 1. The structure diagram of the control system.](image)

3. Hardware System

In the design of hardware system, master controller and its extension board, slave controller, motor drive modules, sensor modules and actuator modules are selected. The control board of slave controller and power module are specially designed based on the control requirements of robot.

- **Master controller and its expansion board:** The master controller, Raspberry Pi 4B, adopts a Broadcom BCM2711 processor with a quad-core CORTEX-A72 and 1.5GHz dominant frequency. Raspberry Pi 4B performs machine learning task more than twice as fast as 3B+, which can better realize image processing and object recognition. Moreover, it is equipped with USB 3.0 data interface, which can achieve faster data transmission. Its extension board extends the Raspberry Pi's IO pin and power module, having IIC, UART interfaces and external devices such as motor drivers, status indicators and buttons. Since Raspberry Pi 4B has only one stable UART, in this control system, this UART is not only used for communication between slave controller and master controller, but also used for controlling serial digital steering gear, whose time-sharing reuse is achieved by using latch 74HC126D.

- **Slave controller and its control board:** The slave controller, STM32f103C8T6, is a 32bit microcontroller with ARM Cortex-M kernel and 72MHz dominant frequency. The control board of slave controller is equipped with reset and clock circuit of STM32, motor driver interface and special sensor interfaces. The control board has three UART communication ports and one IIC communication port.

- **Power module:** The power supply of control system is 11.1 V (3S) lithium battery. Sensors need 5 V power supply, and slave controller needs 3.3 V power supply. In order to meet power supply requirements, power module is designed, in which switch power chip MP1584EN is used to convert 11.1 V to 5 V and linear power chip AMS1117 is used to convert 5 V to 3.3 V.

- **Camera module:** a USB camera with 1.3 megapixel is used, which has good compatibility, driving-free and other advantages. A two-axis pan-tilt is added at the bottom of the camera to expand the monitoring scope of camera.

- **Motor driving module:**
High-powered driving chip L298p is selected as the key element of master controller's motor driver. L298p has two high-power H-bridges, and its output driving current can reach 2A. The motor driver uses 8 high-speed schottky diodes to protect master controller. H-bridge circuit composed of high-powered MOSFET is used as the key element of slave controller's motor driver. The motor driver can drive up to 12A, and can drive a DC motor with a maximum power of 300W. To ensure the safety of slave controller, optical coupling isolation is adopted between the motor driver and slave controller.

- Sensor modules: The long-distance laser range sensor is selected as range sensor, which has an accuracy of 1 mm and a measuring range of 5 cm to 40 m, and sends distance signal to slave controller through UART. Selecting the inertial measurement unit (IMU) MPU6050 as the attitude sensor to measure the acceleration and angular velocity of the robot. Photoelectric encoder is selected as the encoder, which is used to measure speed of robot.

- Actuator modules: The serial digital steering gear with higher precision are used, which has serial port for communication, so master controller needs only one data line to control multiple steering gears. The motor adopts 12V DC motor with a deceleration ratio of 1:30, which can generate a large torque.

4. Software System

The software system design mainly involves program design of the detection and tracking of target and the following motion control of robot. There are three threads working in the master controller, which are infinite loop thread I, infinite loop thread II and a timing task thread. The infinite loop thread I implements target detection and tracking algorithms. The infinite loop thread II executes following decision algorithm. The timing task thread fuses angular velocity and acceleration information given by foot attitude sensor to calculate tilt angle of foot. The slave controller implements periodically the algorithm of motion control for robot.

4.1. Detection and Tracking of Target

Target detection and tracking program needs to be deployed on Raspberry Pi 4B, which includes target detection module and target tracking module. Due to the limitation of processor’s performance and memory, it’s difficult to deploy complex algorithms such as neural networks on it. After several tests, the target detection module adopts multi-scale template matching algorithm, and the target tracking module adopts MedianFlow tracking algorithm based on Lucas-Kanade tracker, which can balance the speed and accuracy well. Block diagram of target detection and tracking algorithm is shown in Fig.2.

![Figure 2. Block diagram of target detection and tracking algorithm.](image-url)

The procedure of target detection and tracking algorithm is described as follows:

- First, the new frame to be detected and the template are preprocessed, and then multi-scale template matching is used on them. If the match is successful, the bounding box information will be obtained. If template matching fails, '"-1" is written to the document and overrides document’s existing content.
- When the template matching is successful, the frame and the bounding box information are used to initialize the target tracking module.
- MedianFlow algorithm is used to predict the bounding box position and size of the current frame based on the position and size of the bounding box of the previous frame. If the tracking is
successful, the center coordinates of the bounding box is written to the document every 0.1 seconds
and overrides document’s existing content.

- Due to the shortcomings of MedianFlow algorithm, improper bounding box is easy to appear when
tracking lasts for a long time, that is, the predicted bounding box is too large or too small. Therefore,
the target detection and tracking program is rerun when the successful tracking time reaches 5
seconds.

- If target is lost in the tracking process, that is, the tracker cannot continue to track the target because
it is out of the view of camera, the target detection and tracking program should be rerun, and '-1'
is written to the document and overrides document’s the existing content.

4.1.1. Target detection algorithm. The camera gets a new frame of 320×240 pixels, this new frame is
converted into grayscale one, and then Canny operator is used for edge detection to obtain the target
image. The template image is processed in the same way and resized in 5 different ratios to get different
scale templates. Then template matching is conducted by using five templates separately, which is
shown in Fig.3.

![Figure 3. Multi-scale template matching.](image)

As shown in Fig.4, the template slides on the target image. \((x, y)\) are the position coordinates of the
template locating at the upper left corner of the target image. The width of the template is \(w\), the height
of the template is \(h\), the width of the target image is \(W\) and the height of the target image is \(H\). For each
pixel point \((x, y)\), equation (1) is used to calculate the correlation coefficient \(R(x, y)\).

\[
R(x, y) = \sum_{x', y'} \left( T'(x', y') \cdot I'(x + x', y + y') \right)
\]  
(1)

\[
T'(x', y') = T(x', y') - \sum_{x''} T(x'' y') \frac{w \times h}{w \times h}
\]  
(2)

\[
I'(x + x', y + y') = I(x + x', y + y') - \sum_{x''} I(x + x'' y + y'') \frac{w \times h}{w \times h}
\]  
(3)

where \(x \in [0, W - w], y \in [0, H - h], x' \in [0, w], y' \in [0, h], x'' \in [0, w], y'' \in [0, h]\). \(T\) represents
the template, \(I\) represents the target image, and \(R(x, y)\) is correlation coefficient. \(R(x, y)\) of each pixel
point constitutes the matching result matrix, whose width and height are \((W - w)\) and \((H - h)\) respectively.
\(R(x, y)\) represents the matching degree of the template and target image at the point \((x, y)\). The degree
of matching is positively correlated with \(R(x, y)\).
Multi-scale template matching refers to using five different scale templates to match the target image, which can obtain five different matching result matrices. Each element in these matrices corresponds to the correlation coefficient of a certain template at a certain pixel point, and the maximum of all these correlation coefficients is taken as the global maximum. When the global maximum is greater than threshold, the matching is considered successful. After several tests, the threshold is set to 6,000,000. The corresponding coordinates \((x, y)\) of this global maximum are taken as the upper left corner coordinates of the initial bounding box whose size is equal to the size of corresponding template. Then initialization of target tracking algorithm is conducted by using the frame to be detected and the initial bounding box. If the global maximum is less than the threshold, another new frame is obtained from camera to conduct multi-scale template matching again.

4.1.2. Target tracking algorithm. The MedianFlow\(^4\)\(^5\) target tracking algorithm proposed by Zdenek Kalal is selected, and the procedure of this algorithm is as follows:

- In the bounding box of the \(t\)th frame, several points are randomly and uniformly generated as feature points.
- Using the Lucas-Kanade optical flow tracker to track the target, it makes grayscale comparison between frame \(t\) and frame \(t + 1\) to estimate the position of feature points in frame \(t + 1\).
- Then the position of these points in frame \(t\) is predicted backward. FB error (forward-backward error) is calculated for each point, and half of the points with smallest FB error are selected as the best tracking points.
- Predict the position and size of bounding box of frame \(t + 1\) according to the coordinates and distance variation of the best tracking points.

FB error’s calculation is shown in Fig.5. Track point \(x_t\) from frame \(I_t\) to frame \(I_{t+k}\), and then predict \(\hat{x}_t\) from frame \(I_{t+k}\) to frame \(I_t\). \(\hat{x}_t\) is the corresponding point in frame \(t\). The Euclidean distance between \(\hat{x}_t\) and \(x_t\) is the FB error of the tracker at frame \(t\).

Due to the limitation of MedianFlow algorithm, after several seconds of continuous successfully tracking, the size of bounding box will be not suitable. Therefore, the program is set as follows: if the continuous successful tracking time reaches 5 seconds, rerun the target detection and tracking program. Once target is lost during the tracking process, the target detection and tracking program will be rerun.

4.2. Following Motion Control for Robot

4.2.1. Following decision algorithm for robot. General speaking, speed and turning angle of the robot should be determined according to the distance and angle of robot relative to target. In order to obtain recognition result of target more conveniently, following decision algorithm for robot is implemented in master controller. The procedure of the following decision algorithm is as follows, whose block diagram is shown in Fig.6.

- Tracking information of target is parsed from the TXT file. If the information is ’-1 ’, two-axis pan-tilt is controlled to turn left or right for expanding monitoring range of camera, otherwise, central coordinates of bounding box of tracking target are read, and deviations between these coordinates and image center’s coordinates are used to control two-axis pan-tilt, so as to keep the tracking target at the center of image.
- Expected turning angle of the robot is calculated by using rotation angle of two-axis pan-tilt, which is the set value of turning angle PID controller.
- The angular velocity and acceleration given by IMU of robot’s body are fused by Kalman Filtering algorithm to calculate upright angle of robot body.
- According to the rotation angle of two-axis pan-tilt and upright angle of robot body, the included angle between the laser range sensor and horizontal position is calculated. This included angle and the distance measured by the laser range sensor is fused to obtain distance between robot and target based on cosine theorem.
- Finally, expected speed of the robot is determined according to the distance between robot and target, and this speed is taken as set value of speed PID controller.
4.2.2. Motion control algorithm for robot. When controlling motion of robot, it is necessary to perform its upright angle control, speed control, turning angle control and adjustment to center of gravity. In order to ensure the real-time control of robot, motion control of the robot is realized by slave controller. The block diagram of the motion control algorithm is shown in Fig.7. The procedure of this algorithm is as follows.

- Using Kalman Filtering algorithm, upright angle of robot’s body is calculated based on angular velocity and acceleration given by body attitude sensor, and tilt angle of foot is calculated based on angular velocity and acceleration given by foot attitude sensor.
- When robot is on uphill or downhill, IMU of foot is in a long-time tilt state. If tilt angle of the IMU is more than 5° or less than -5° during consecutive 5 sampling period, robot is considered to be on slope, and the steering gear of the ankle is controlled according to the tilt angle to achieve forward and backward tilt of robot so that the robot can keep the center of gravity in balance.
- Upright angle control and speed control adopt cascade PID controller, turning angle control adopts PID controller[6]. The sum of the output of speed PID controller and upright angle error is input to upright angle PID controller, and the sum of the output of the upright angle PID controller and output of the turning angle PID controller is used as control quantity of motor (PWM). Robot is supported by four wheels, when upright angle of robot body being within -8° to 8° (expected upright angle = 0), robot can remain upright self-balance, so only when upright angle error exceeds this balance range, upright angle error will be introduced to upright angle control. The control block diagram is shown in Fig.8.
5. Conclusion
In this paper, the control system for wheeled humanoid following robot is summarized. In this control system, visual method is adopted to recognize target. Kalman Filtering algorithm is used to calculate angle of robot. Cascade PID is introduced to control robot's upright angle, speed and turning angle. Multi-scale template matching algorithm and Median-flow algorithm are adopted in target recognition. Because Median-flow algorithm is imperfect, after a few seconds of tracking, the predicted target’s bounding box is not improper. In order to overcome this defect, the multi-scale template matching and tracking algorithm is implemented periodically. Experiment results show that this low-cost wheeled humanoid robot has good self-balancing stability, anti-interference performance and following stability in the process of tracking the target.

Target recognition algorithm based on neural network is more accurate and efficient. Raspberry Pi processor cannot implement neural network algorithm directly due to its performance limitations. If the Raspberry Pi is equipped with a neural computing stick, the neural network algorithm can be used for target recognition. In addition, more stable self-balancing control of robot can be achieved by controlling its joints instead of the robot's speed. Of course, to achieve this goal, the forward kinematic equation and inverse kinematic equation of the robot need to be established.

Acknowledgments
This work is supported by Beijing Institute of Fashion Technology (Grant No. NHFZ20190152/004).

References
[1] Qi Liping. 2015. Research of object perception based on vision for home service robot[D]. Hebei University of Technology.
[2] Liu Hao. 2019. Research on robot target tracking system based on vision[D]. Inner Mongolia University of Science and Technology.
[3] Chen Zihan, Du Yuxiao, Li Buheng, Lai Yufan, Ke Jiaying, Zeng Tianrui. 2018. Humanoid robot system based on dual controllers[J]. Automation and Information Engineering, 39(05), p38-42.
[4] Zdenek Kalal, Krystian Mikolajczyk, Jiri Matas. 2011. Tracking-Learning-Detection[J]. IEEE Trans Pattern Anal Mach Intell, 34(7):1409-1422.
[5] Kalal Z, Mikolajczyk K, Matas J. 2010. Forward-backward error: automatic detection of tracking failures[C]. 20th International Conference on Pattern Recognition, ICPR 2010, Istanbul, Turkey, 23-26 August 2010. IEEE, 2010.
[6] Du Haowen. 2019. Research and realization of the tracking scheme of the two-wheel self-balanced intelligent vehicle[D]. Guangxi Normal University.