Self-balancing Robot Design and Implementation Based on Machine Vision

Yingnian Wu(✉) and Xinli Shi

School of Automation, Beijing Key Laboratory of High Dynamic Navigation Technology, Beijing Information Science & Technology University, Beijing, China
wuyingnian@bistu.edu.cn

Abstract. Self-balancing robot based on machine vision with two wheels sometimes can be more flexible and saving space than four wheels. We present design scheme and experiment implement of self-balancing robot based on machine vision in the paper. The robot can get road information by CCD camera, then process the image by algorithm to get the road path information and the robot can safely and reliably auto-drive along the road. Based on machine vision the self-balancing robot does not need person to control it, and it can reduce the human disturbance which can greatly improve the efficiency and safety of the system. The system needs to deal with lots of tasks under CPU resource limitation. We have considered the task scheduling under CPU resource limitation through experiment and theory prove. It is proved that the system has good intelligence and anti-interference ability by experiments and competition.

Keywords: Self-balancing robot · Machine vision · Task scheduling

1 Introduction

With the development of new information technology, more and more robots have been used in factory. Self-balancing robot based on machine vision with two wheels sometimes can be more flexible and saving space than four wheels. It has the function of automatic driving, and automatic identification of road information. It can get the environment around itself based on the camera. There are black lines in both edges of the road. The robot can safely and reliably auto-drive along the road. Based on machine vision the self-balancing robot does not need person to control it, and it can reduce the human disturbance which can greatly improve the efficiency and safety of the system [1–4].

2 System Model and Stability Discussion

Figure 1 is the real system of the self-balancing robot. We can take the robot as a physical pendulum model which centroid weight is m and pendulum length is L, and take the motor and its driver system as coil spring model. We can discuss the pendulum stability from the perspective of the potential energy curve.

L. Zhang et al. (Eds.): AsiaSim 2016/SCS AutumnSim 2016, Part II, CCIS 644, pp. 29–37, 2016.
DOI: 10.1007/978-981-10-2666-9_4

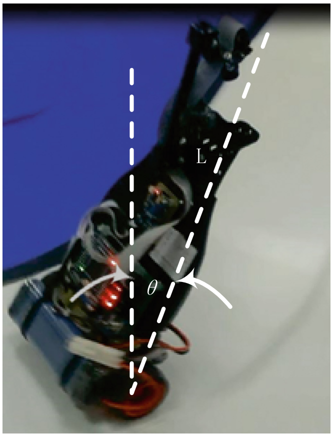

Fig. 1. Real system of the self-balancing robot.

Suppose the system satisfying Hooke's law, which assumes that the restoring force of the motor and the tilt angle is proportional to the actual situation. In the real system, when there is a deviation between the setting angle and the real angle of the robot, the motor will timely correct the angle. So we assume the angel between setting angel and real angel of the robot is less than 5 degree. The restoring force of the motor is proportional linear relationship to the inclination angle.

Set the motor torque of the robot inverted pendulum model:

$$\mathrm{L}_b = -k\theta \tag{1}$$

The elastic potential energy is

$$\mathrm{E}_{pb} = \int_0^\theta k\theta \mathrm{d}\theta = \frac{1}{2}k\theta^2 \tag{2}$$

The gravitational potential energy is

$$\mathrm{E}_{pg} = mgL(\cos\theta - 1) \tag{3}$$

The total potential energy is

$$E_p = \frac{1}{2}k\theta^2 + mgL(cos\theta - 1) \tag{4}$$

Balance position is corresponding to the extreme value of the potential energy curve, so set $E_p = U(\theta)$, we can get the possible existing equilibrium position by derivative of θ.

$$\frac{\mathrm{d}U}{\mathrm{d}\theta} = k\theta - mgL\sin\theta = mgL\left(\frac{k}{mgL}\theta - \sin\theta\right) \tag{5}$$

Set $\frac{\mathrm{d}U}{\mathrm{d}\theta} = 0$, there exist 3 cases, case 1: $\frac{k}{mgL} > 1$, only if $\theta=0$, there exists a stationary point; case2:$\frac{k}{mgL} < 1$, there exist 3 stationary points, $\theta=0$ and $\theta = \pm\,\theta_0$. case3: The critical state between the case 1 and case 2.

The stability of the equilibrium position is decided by $\frac{\mathrm{d}^2U}{\mathrm{d}\theta^2}$.

$$\frac{\mathrm{d}^2U}{\mathrm{d}\theta^2} = k - mgLcos\theta \tag{6}$$

(1) When $\frac{k}{mgL} > 1$ and $\theta=0$, $\frac{\mathrm{d}^2U}{\mathrm{d}\theta^2} > 0$, $E_p = U(\theta)$ exists minimum value, equilibrium is stable.

(2) When $\frac{k}{mgL} < 1$ and $\theta=0$, $\frac{\mathrm{d}^2U}{\mathrm{d}\theta^2} < 0$, $E_p = U(\theta)$ is corresponding to the potential energy maximum value, equilibrium is unstable.

(3) When $\frac{k}{mgL} < 1$ and $\theta = \pm\,\theta_0$, $\frac{\mathrm{d}^2U}{\mathrm{d}\theta^2} < 0$, which can prove, equilibrium is stable [5].

3 Hardware System Design

There are 6 modules in the self-balancing robot based on machine vision systems: micro-controller module, wireless communication module, machine vision module, inertial sensing module, power module, and motor drive & control module (Fig. 2).

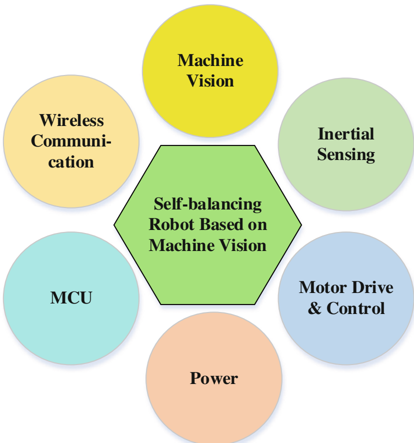

Fig. 2. Robot system scheme

3.1 Power and Motor Drive & Control Module

The main functionality of the power module is to supply power for the micro controller, all sensors and motor driver modules. The robot uses Ni-Cr battery for power supply, which has good specialty for big electricity discharging. The specified voltage for the battery is 7.2 V, which can be up to 8.4 V with fully charging. The deceasing voltage under load is small, which makes benefit for the robot to move stably.

The CCD camera, micro controller, encoder, blue tooth, accelerometer and gyroscope module are all powered at 5 V, hence the regulator module is only made up of LM2940 regulator chip, which is relatively simple and has small effect on the gravity center of the robot.

The motor drive & control module is made up of BTS7970B double circuit using light coupling, so that it can be isolated from the power supply on CPU and effectively protect the CPU. The drive circuit has a small size and leads to good driving ability and effective braking.

3.2 Micro-Controller Module

The micro-controller is the brain of the intelligent robot. It controls the robot to perform various actions as pre-settings, receives the information gathered by all sensors, processes all received information, and transfers the processed information to corresponding actuators.

The robot uses the 16 bit MC9S12XS128 MCU as its control core. The main frequency is 16 MHz, which can be frequency doubling increased to 96 MHz by the phase locked loop inside the MCU. The interrupt of MCU has seven priorities and its kernel supports priority dispatching, which can access the whole memory space up to 8 MB.

MC9S12XS128 is a 16 bits MCU, which has a limitation on processing speed and resource comparing to 32 bits MCU. We optimized the program and system during research and applied simple and effective methods to implement expected functions.

3.3 Wireless Communication Module

The main function of wireless communication module is to transmit the data from MCU to the superior workstation. The blue tooth is the protocol for short distance communication. Generally, the communication distance is about 10 meters at the open area and the communicating speed gets lower and lower upon longer distance. The system applies the blue tooth to transmit the data from the robot to the superior workstation so that debugging and status monitoring can be performed.

3.4 Machine Vision Module

The robot applies CCD camera to get the road information during moving. There are black border lines on both sides of the road. According to the road border lines information gathered by the camera, the robot can move automatically between the border lines. The CCD camera OV7620MVA applies self-adoption power supply at 5 V/3.3 V, no need to increase the voltage to 12 V, which makes the regulator circuit simple, the regulator module lighter, and the gravity center lower. Also, the camera uses 2.8 mm/95 wide lucid lens to ensure a high quality for the images.

3.5 Inertial Sensing Module

The robot controls the motor speed by gathering and processing the data from inertia sensing module, hence controls its balance and moving speed. The inertia sensing module consists of accelerometer, gyroscope and encoder.

The accelerometer and gyroscope module uses MMA7361 and ENC03 chips, a triple axis accelerometer and a single axis gyroscope module. This module applies a hardware Kalman filter. The Kalman filter can merge the signals from accelerometer and gyroscope.

The encoder is 512 lines triple axis MINI incremental rotating encoder with small size. When the disk with optics graphic pattern and rotating axle rotate at the same time, the corresponding light getting through the two aperture will appear as passing through and intercepting status, hence the angle displacement can be transformed to corresponding pulse signal [6], and then figure out the moving speed of the robot.

4 Software System Design

The software system of the robot consists of main function module, one-millisecond interrupt system module and video field interrupt system module.

The main function module is the frame of the whole program and the base for both one-millisecond interrupt system and field interrupt system module. The one-millisecond interrupt module is a program based on MC9S12XS128 one-millisecond interrupt hardware, and millisecond interrupt will occur every millisecond. The field interrupt system is used to process image information from the CCD camera. During the idle time for CPU to process the main program, it will process the one-millisecond interrupt program or field interrupt program at regular intervals. If there is any conflict for the two programs, process the field interrupt program as higher priority.

4.1 Main Function Module

Figure 3 illustrates how the main program works. It starts to wait for interrupt right after initialization. MC9S12XS128 totally has 55 interrupt with different priorities, and the one with highest priority is reset interrupt. The robot only uses two different priority interrupt, i.e., field interrupt and one-millisecond interrupt. That is, the core board will run the field interrupt, or one-millisecond interrupt, or stay idle right after the initialization.

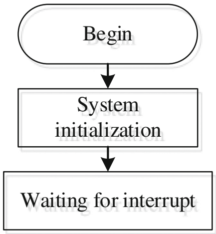

Fig. 3. Main function module diagram

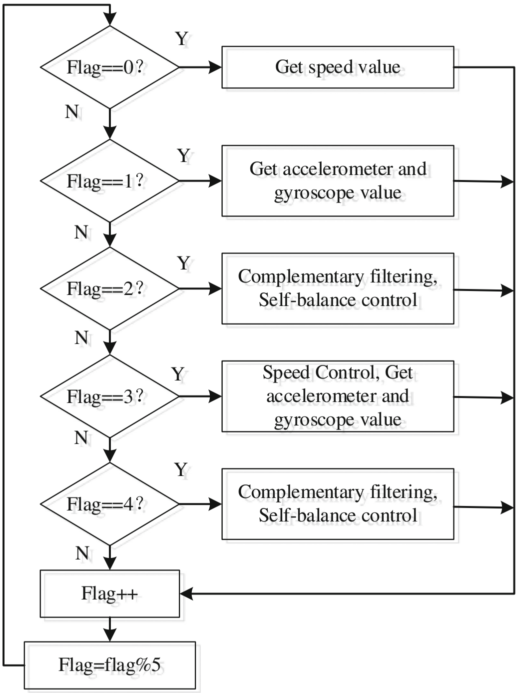

Fig. 4. Millisecond interrupt diagram

4.2 One-Millisecond Interrupt Module

Figure 4 illustrates the one-millisecond interrupt module. One-millisecond interrupt means one interrupt at every millisecond, that is, it will process interrupt regardless of whether the interrupted task is completed or not. It requires that the tasks assigned during each interrupt could not last longer than one millisecond, otherwise, the CPU could not dispatch the tasks well. Flag is the flag specified for millisecond interrupt. Getting the robot speed, the value of accelerometer and gyroscope, complementary filter, self-balance control and speed control are all corresponding sub functions.

4.3 Field Interrupt Module

The field interrupt has higher priority than the millisecond interrupt. The module consists of function to capture video and process the image and function for direction control.

4.3.1 Video Capture and Image Process Function

Firstly, the camera will capture one frame image of the road track. Then the CPU will process the 128 pixel point on the first line of the frame image and make it binaryzation. The reason to get the value of the first line is because we found it can make the robot perspective with less disturbance point from many tests on the image controlling. After obtaining the 128 binaryzation data, it will retrieve the data of this line from both left and right sides to find the junction of the black line on left and right sides and the white track, and return a value. After that, it will figure out if the track is straight or curve even sharp curve according to the range of returned value, then get a middle value corresponding to different processing algorithm. Moreover, to remedy the deviation caused by getting a single line, the algorithm applies *fake multiple lines processing* method, that is, to store the middle value, superpose multiple middle values with different proportions according to straight or curve track, and get the final middle value. All these proportion values are the optimized values after many tests.

4.3.2 Direction Control Function

The direction control function gets the middle value from the Video capture and image process function. Then it figures out whether current track is straight or curve even sharp curve by checking if the difference is big between current middle value with the next one. By combining the middle value with PID parameters and adjusting the PID parameters, it can adjust the stability of the direction control function.

4.4 The Task Scheduling Under CPU Resource Limitation

To save the cost, the system applies a lower main frequency CPU. As the system is composed of multiple sub systems, we must arrange the task scheduling reasonably to ensure the system running reliably. So we measured the actual tasks execution time by experiments after system designing. We measured the execution time of each function and figured out whether a CPU static task scheduling can be performed by setting a pin to high level at the beginning and setting it to low level at the end of the function, and using dual-trace oscilloscope to measure the high level time of each function. The actual execution time of each task was obtained by averaging the ten times measurement results.

According to the setting, the polling interval for above five tasks is 5 ms. Above tasks will be executed within the millisecond interrupt function periodically by sequence. The field interrupt has higher priority than millisecond interrupt, so the CPU will pause the millisecond interrupt immediately and start the field interrupt if the field interrupt comes. It costs less than 7 ms to read one image data, and the interval between reading each frame image is 16.67 ms (Table 1).

Table 1. Task execution time

Task no.	Measure value(μs)											Average (μs)
0	Speed_Read	20	19	20	18	22	20	23	17	21	19	19.9
1	Get_ADResult	165	152	148	162	150	156	158	142	150	164	154.7
2	Datehandle	18	23	15	19	20	22	16	21	17	20	19.1
	Stand_Control	162	149	154	157	142	160	162	145	152	155	153.8
3	Speed_Control	13	17	22	25	18	15	21	24	16	18	18.9
	Get_ADResult	157	162	150	162	158	165	148	152	160	156	157.0
	SpeeedControlOutput	20	15	22	26	18	25	16	22	15	19	19.8
4	Datehandle	16	14	18	22	25	20	15	24	25	17	19.6
	Stand_Control	160	155	162	164	158	148	142	145	152	152	153.8

There are 6 tasks including 5 polling tasks and one field interrupt task. According to optimal RM task scheduling theorem [7, 8], the occupancy of the tasks is,

$$
\begin{aligned}
U &= \sum_{i=1}^{n} \frac{c_i}{h_i} \\
&= \left(\frac{19.9}{5} + \frac{154.7}{5} + \frac{19.1 + 153.8 + 18.9}{5} + \frac{157 + 19.8}{5} + \frac{19.6 + 153.8}{5}\right) \times 10^{-3} + \frac{7}{16.67} \\
&= 0.5632 < n(2^{\frac{1}{n}} - 1) = 6 \times (2^{\frac{1}{6}} - 1) = 0.7348
\end{aligned}
$$

The system tasks can be scheduled.

To realize the goal as much as possible and avoid the impact among internal interrupts in MC9S12XS128, we only transformed 182 columns per line during image data transformation. Although to obtain the road path information by gathering only one line data will lead to more errors, we have to keep doing like that as we found the robot cannot self-balance when reading more than one lines of image data. Fortunately, we can remedy it by increasing fault tolerance and adjusting PID parameters during direction control.

5 Conclusion

The auto-balanced robot based on machine vision has been designed and experimented in the paper. Based on machine vision the self-balancing robot does not need person to control it, and can reduce the human disturbance which greatly improves the efficiency and safety of the system. The system needs to deal with lots of tasks such as getting the robot speed, getting the value of accelerometer and gyroscope, complementary filter, self-balance control and speed control, capture video and process the image and direction control etc. We have considered the task scheduling under CPU resource limitation through experiment and theory prove. The works in the paper has attended the national robot competition based on machine vision won the second prize. It is proved that the system has good intelligence and anti-interference ability by experiments and competition.

Acknowledgments. This work is supported by the BISTU Graduate Education Quality Project (5111623305),BISTU Course Construction and Teaching Reform Project (2014KG22, 2015KGZD06, 2016JGYB15, 2014JG08).

References

1. Umar, F., Usman, M.A., Athar, H., et al.: Design and implementation of a fuzzy logic controller for two wheeled self-balancing robot. Adv. Mater. Res.: MEMS, NANO and Smart Syst. **403-408**, 4918–4925 (2012)
2. Osama, J., Mohsin, J., et al.: Modelling control of a two-wheeled self-balancing robot. In: International Conference Proceedings on Robotics and Emerging Allied Technologies in Engineering, pp. 191–199 (2014)
3. Yuli, C., Yikai, S., et al.: Research on two wheeled self-balanced robot based on variable universe fuzzy PID control. Comput. Simul. **30**(2), 347–350 (2013)
4. Wenjian, L., Hang, Z., et al.: Design and implementation of control system for two-wheeled self-balancing robot. J. Electron. Meas. Instrum. **27**(8), 750–759 (2013)
5. Shouxian, S.: Stability and bistability in physical systems (I). Phys. Eng. **12**(03), 1–5,10 (2002)
6. Wen, M., Hong, Y., et al.: Application of incremental encoder in impulse measurement with compound pendulum. Chin. J. Sci. Instrum. **28**(1), 140–144 (2007)
7. Shouping, G., Wei, Z., et al.: Networked Control Systems and Application. Publishing House of Electronics Industry, Beijing (2008)
8. Wu, Y., Song, X., Gong, G.: Real-time load balancing scheduling algorithm for periodic simulation models. Simul. Model. Pract. Theor. **52**(1), 123–134 (2015)