*Applied Mechanics and Materials*
*ISSN: 1662-7482, Vols. 602-605, pp 989-992*
*doi:10.4028/www.scientific.net/AMM.602-605.989*
*© 2014 Trans Tech Publications, Switzerland*

*Submitted: 2014-05-31*
*Accepted: 2014-06-11*
*Online: 2014-08-11*

# Software design of micromouse motion control

Zhang Haoming[1, a], PEH lian Soon[2,b] , Wang Yinghai [3,c]

[1] Department of Electrical and Information Engineering, Tongling University, Anhui Province, China

[2]SHINING SUNSHINES GLOBAL PTE LTD, Singapore

[3] School of Electrical and Information Engineering, SIPIVT, Jiangsu Province, China

[a]zhm20060616@163.com, [b] bai_ls@yahoo.com, [c] yin287362@163.com

**Keywords:** Micromouse, Maze, PD control, Motion profile generator

**Abstract.** Micromouse is a small autonomous robot, which must navigate itself through an unknown maze from the start to the destination.One of the main challenges for micromouse is to control its two motors, which are used to control micromouse's motion in solving the maze. Detailed design of an improved PD controller is given,which has to update the velocity and acceleration values of the motors when it wants to control the robot. Motion profile generator and motion command execution routine is designed to output a smooth motion for the robots.

## Introduction

Micromouse championship is an international robotics competition, which is an event where small robot micromouse solves a 16x16 maze[1-3],as shown in figure 1.

Fig.1 Micromouse solves maze

Micromouse championship began in 1950. Events are held worldwide, and now are most popular in the UK, US, Japan, Singapore, India and South Korea.

The micromouse is completely autonomous find their way from a predetermined starting position to the central area of the maze unaided. For the system, it is mainly controlled by a MCU and a FPGA. All of the calculations and decision makings will be executed by the MCU, while the FPGA handles the pulse-width modulation (PWM) of motors. The motors are connected to the dual full bridge driver which is built in the electronic system. Under the help of sensors, controller sends control signal to two brushed DC motors to make it run[4-8].

In order to control the DC motors, the author implemented a proportional-derivative (PD) controller in the system. A PD controller is a control loop feedback mechanism most commonly and widely used in industrial control systems, as shown in figure 2.

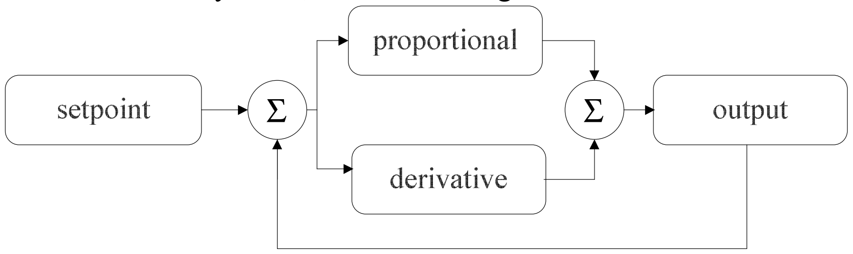

Fig.2 The block diagram of a PD controller

## The improved PD controller

The PD controller calculates the "error" value as the difference between the actual process variable (the motor position) and the desired set-point (the calculated position).The controller will try to minimize the error by adjusting the process control inputs, as shown in figure 3. After calculating the error, the controller will decide when to change the motor position and by how much. This is the part of proportional control. In the interest of achieving a gradual convergence at the desired position, the controller will have to damp the anticipated future oscillations and it may temper the adjustments. This is the part of derivative control.

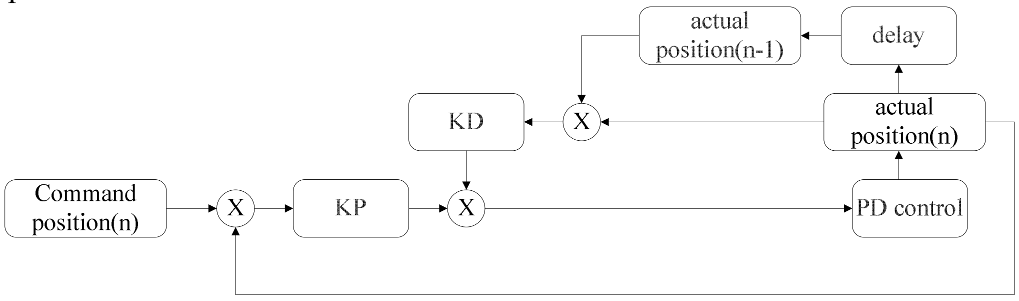

Fig.3 The block diagram of the implemented PD controller

For the Proportional term, the PD controller will capture the actual position of motor by reading the encoder count from the quadrature encoder which is implemented inside the FPGA and the controller will calculate the difference between the actual position and the command position which is the desired setpoint for the motor control process and calculated by the profile generator. The difference is called ‘Err1’ in the PD routine. The proportional term makes a change to the output that is proportional to the current error value. The proportional response can be adjusted by multiplying the error by a constant Kp, called the proportional gain.

For the derivative term, the PD controller will calculate the difference between the actual position and the previous actual position which is stored inside the system before capturing the new actual position. The derivative of the process error is calculated by determining the slope of the error over time and multiplying this rate of change by the derivative gain KD. The magnitude of the contribution of the derivative term to the overall control action is termed the derivative gain, KD.

In order to improve the performance of PD controller, the system is improved by adding another derivative control term, as shown in figure 4.

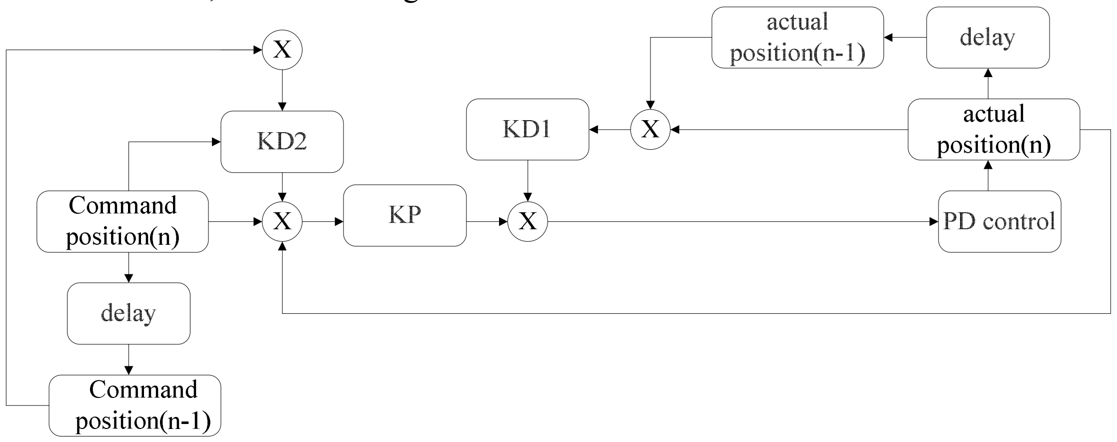

Fig.4 The block diagram of the improved PD controller

The improved PD controller will calculate the difference between the current command position and the previous command position which is stored inside the system before updating the new command position. The magnitude of the contribution of this second derivative term to the overall control action is termed the second derivative gain, KD2.

## Motion profile generator

A motion profile generator is designed to output a smooth motion for the robot. The system has to update the velocity and acceleration values of the motors when it wants to control the robot, then the motion profile generator will try to change the actual velocities according to the desired values. Instead of dumping the values which is using the ramp profile and will cause the slippage, the profile generator will accelerate the motors using the partial trapezoidal profile, as shown in figure 5.

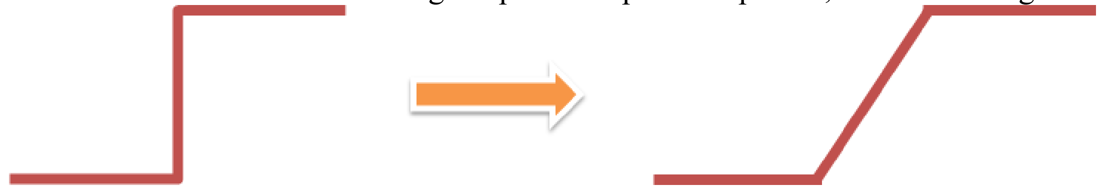

Fig.5 The ramp profile vs partial trapezoidal profile

The profile generator will keep updating the command position data which will be used in the PD control calculation. The auto-navigation module also will update the command position data when it has to navigate the robot back to the correct moving direction. The reason why the auto-navigation module directly controls the command position data instead of changing the accelerations and velocities is to response faster when the robot is going to crash with the obstacles.

Figure 6 shows that the profile generator will keep comparing the actual velocities with the system defined velocities.

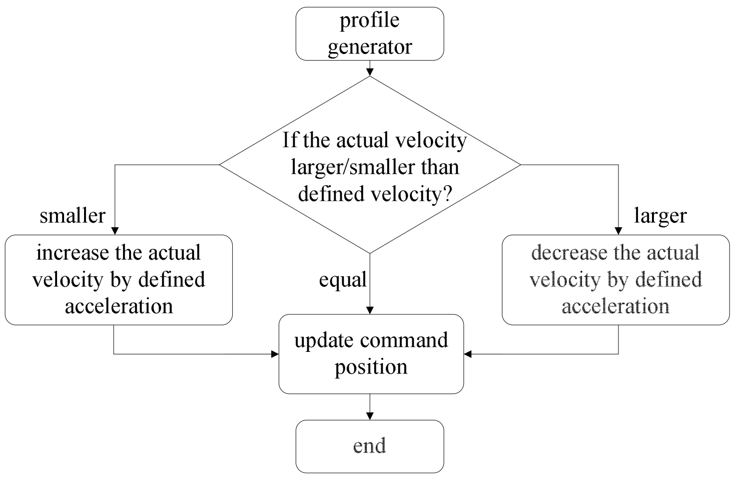
profile generator
If the actual velocity larger/smaller than defined velocity?
smaller
larger
equal
increase the actual velocity by defined acceleration
decrease the actual velocity by defined acceleration
update command position
end

Fig.6 The flowchart of motion profile generator

If the actual velocity is larger than the defined velocity, then the actual velocity will be decreased by the defined acceleration value. If it is a ‘smaller’ case, then it will be increased by the defined acceleration value. After the update of velocity, then the profile generator will update the command position data according to the velocities.

And all the navigations which will be discussed in the later part are actually running right after the update of the command position. The navigations will have to control the motors directly instead of changing the velocities in order to make the robot responses faster when it faces the obstacles.

## Motion command execution

A motion command execution routine is developed to control the robot by motion commands.The routine will extract one command and check whether it is possible to follow before calling the running or turning modules.

As shown in figure 7, if the robot finds that there is a wall blocking the cell it intends to move into, then the motion command execution will request for another solving solution. The motion command execution will not stop the robot immediately while a new list of motion commands is being generated. The reason why it is not going to stop the robot immediately is to save the travelling time and it will stop the robot if the new command list has not been generated and it is too near to the obstacle or the new next command is a ‘stop’ command.

After the checking, the robot will call the running or turning modules if there is no obstacle blocking the way.

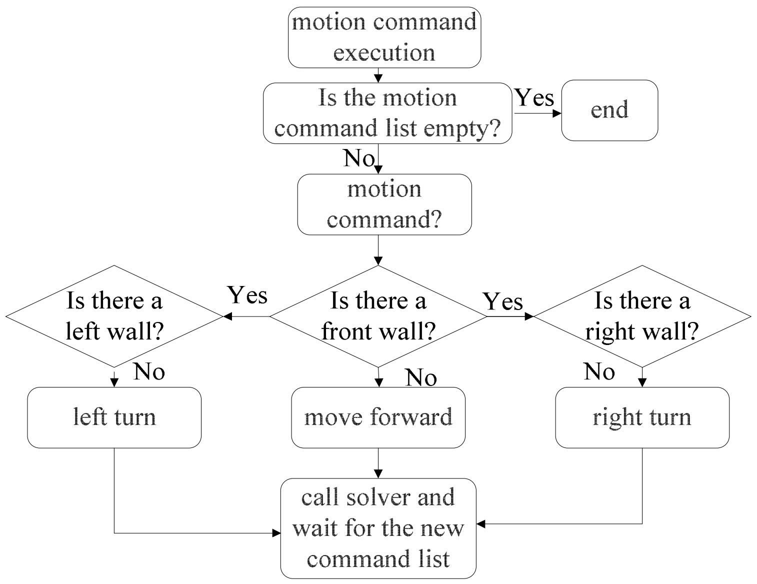

Fig.7 The flowchart of motion command execution

## Summary

One of  the main challenges for micromouse is to control its two motors when solves the maze. when the controller wants to control the robot, which has to update the velocity and acceleration values of the motors. Detailed design of an improved PD controller is given. A motion profile generator is designed to output a smooth motion for the robot.

## Acknowledgement

It is a project supported by basic research programs of Suzhou science and Technology Department - industrial application part (SYG201327) and 2012 Innovation Project of JiangSu Province.

## References

[1] Su,Juing-Huei; Lee, Chyi-Shyong; Huang, Hsin-Hsiung, etc. A micromouse kit for teaching autonomous mobile robots[J], International Journal of Electrical Engineering Education, 2011,Vol.48(2):188-201

[2] Mishra Swati, Bande Pankaj. Maze Solving Algorithms for Micro Mouse[C]. Proceedings of the 2008 IEEE International Conference on Signal Image Technology and Internet Based Systems.2008, 11.

[3] Kiblcr S G,Ilauer A E, Giesscl D S, etc. IEEE Micromouse for mechatronics research and education[C],Mechalronics (1CM), 2011 IEEE International Conference: 887-892.

[4] Po-Jen Cheng,Chin-Hsing Cheng,Chun-Cheng Tung. Design of DC motor's torque control using DSP[J]. Journal of Information and Optimization Sciences,2009,Vol.30(6):1197-1207

[5] Zhang Haoming, PEH lian Soon,Wang Yinghai. Two axes micromouse PWM generator based on FPGA[J].Applied Mechanics and Materials Vols. 496-500:1674-1680.

[6] Zhang Haoming, PEH lian Soon,Wang Yinghai. Design of a Edubot--Micromouse based on MCU+FPGA[J].Applied Mechanics and Materials Vols. 496-500:1664-1669.

[7] Jiang Xiong,Ren Hua-long,Ma Zhong-li. Research on How ARM-based Micromouse Gets Out of Maze[J], Modern Electronics Technique , 2011(08):14-16.

[8] TONDR A D, HALL Drew. The inception of chedda: a detailed design and analysis of micromouse [ D] . Las Vegas: University o f Nevada, 2004.