Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

# A Hands-on Laboratory for Autonomous Mobile Robot Design Courses

**Chyi-Shyong Lee*, Juing-Huei Su*, Kuo-En Lin*, Jia-Hao Chang*, Ming-Hsien Chiu*, and Gu-Hong Lin***

*Lunghwa University of Science and Technology, Taoyuan county, 333
Taiwan (Tel: 886-8209-3211 ext. 5400; e-mail: suhu@mail.lhu.edu.tw).

**Abstract:** This paper presents a hands-on laboratory for autonomous mobile robot design courses, which will be offered to the third or fourth year students of the department of electronic engineering in the Lunghwa University of Science and Technology. The aim of the hands-on laboratory is to introduce to the students the basic implementation issues of autonomous mobile robots from a practical point of view. Therefore, a simple line following robot with automatic control algorithms to detect the line position and to steer the motors, and a micro-mouse capable of solving a given maze are devised for the hands-on laboratory. These two autonomous mobile robot kits based on the dsPIC microcontroller are used to help students learn easily both the hardware and software implementation issues. SIMULINK behaviour models are also developed to help students quickly grasp the working principles of these two autonomous mobile robots. All the necessary software is given at no cost to the students during the class.

## 1. INTRODUCTION

Robots are attracting more and more people's attention recently (NASA, 2007), especially when Sony and iRobot announced the biped humanoid QRIO robot (L. Geppert, 2004) and Roomba vacuum cleaning robot (iRobot cooperation, 2007), respecively. Robots are mechatronic engineering products, capable of acting autonomously implementing assigned behaviors in various physical environments. The widespread use of robots in many areas makes the fundamental understanding of them a necessity for many electronic system design engineers (Qu and Wu, 2006). Unfortunately, learning the design philosophy of mobile robots is interesting but difficult, because it includes many areas of knowledge, e.g., the mechanics and electronics, linear and nonlinear vehicle control system theory, and software programming of micro-controllers, etc. Teaching the autonomous mobile robot design course is a challenging undertaking because one can not assume that all students enrolled in the class have solid prerequisite knowledge in so many areas. Therefore, to speed up the learning process by using the concept of constructionism (A. Thurmond, 1999), the applications of learning-by-doing hands-on laboratory to help students get acquainted with the design and implementation of mobile robots is inevitable.

The hands-on laboratory devised in this paper for learning implementation issues about autonomous mobile robots includes the simple line following robot, and the micro-mouse capable of solving a given maze (IEEE CSUC student chapter, 2004). The project is also supported free of charge those necessary microcontrollers by the local branch of Microchip Technology Inc. The hands-on laboratory is designed for students who have preliminary knowledge about microcontrollers and programming skills. This is why the course will be offered to the third or fourth year students. Students would learn what the essential parts are for the line following robot and the micro-mouse and their functions. To save time, the students would assemble the two robots by using the kits developed in the Lunghwa University of Science and Technology. The following sections would describe in details both the hardware, software and what students can learn by doing the hands-on laboratory about the knowledge involved in implementing the mobile robots.

## 2. OVERVIEW OF THE LINE FOLLOWING ROBOT

### 2.1 Hardware

The line following robot in the hands on laboratory for autonomous robot design courses whose block diagram is shown in figure 1 is a single board computer and it costs about USD$76 (see table 1 for cost analysis).

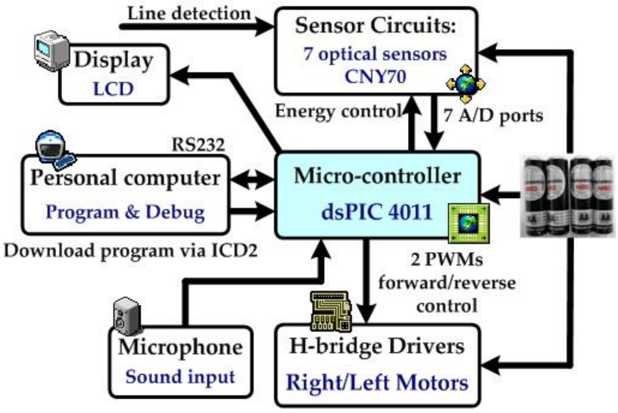


Fig. 1. The block diagram of the line following robot

The line following robot shown in figure 2 consists of a microcontroller dsPIC30F4011 (Microchip Technology, 2007), dual H-bridge dc-motor drivers, an RS232 port to communicate with a PC, an LCD display for debugging, an in system programming (ISP) port for downloading programs from a PC to the 48K byte flash memory, 9 A/D ports for accessing the sound signal, the reflective optical sensor circuit signals, and a microphone to detect the sound of a whistle in a racing contest.

978-1-1234-7890-2/08/$20.00 © 2008 IFAC

10.3182/20080706-5-KR-1001.1158

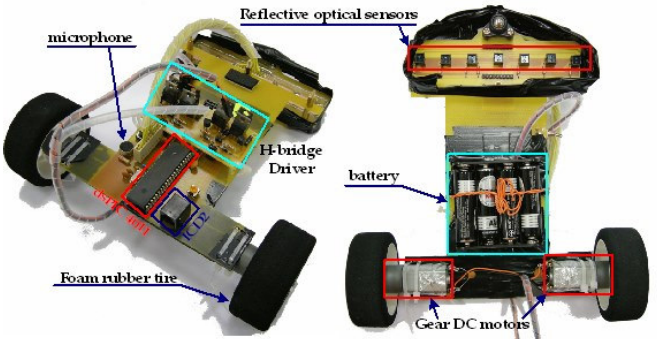


Fig.2 The hardware of the line following robot

Table 1. Cost analysis of the line following robot

| Components and subsystems | $USD |
|---|---|
| dsPIC30F411, An LCD display | 12 |
| 7 Reflective optical sensors (CNY70) | 11 |
| ISP, Power circuit and batteries | 17 |
| Dual H-bridge driver circuits | 6 |
| 2 dc gear motors + crutches + tyres | 30 |
| **Total** | **76** |

To teach students how to use energy in a more efficient way, the working frequency of the micro-controller is lowered down and the current flows through the 7 reflective optical sensors is controlled via a transistor switch shown in figure 3 to save energy. Each reflective optical sensor circuit consumes about 40mA when the VDD is around 6 V.

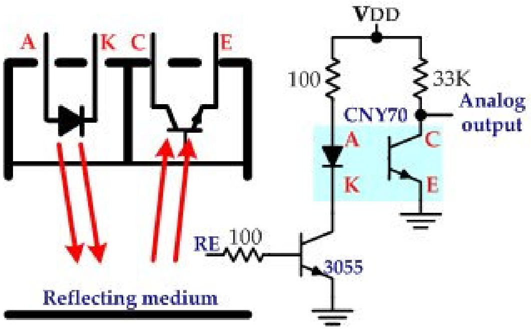


Fig.3 The operation of the reflective optical sensor and the corresponding energy saving circuit

The driving wheels of the line following robot are made with IG-16GM gear motors (Shayang, 2000) and rubber tires. The translational and rotational speed of the robot is controlled by the right and left PWM controlled powered wheels. If the length between the two powered wheels is $L$, the translational and rotational speeds $v$, $\omega$ are determined via the following formulas:

$$v=\frac{v_L+v_R}{2},\quad \omega=\frac{v_R-v_L}{2L}, \tag{1}$$

where $v_L$ and $v_R$ stand for the linear velocities of the left and right powered wheels.

### 2.2 *Normalization of sensor outputs*

To detect the line to be followed, most people use two or more reflective optical sensors, and the outputs of these sensors are fed to comparators with predefined threshold values to get the corresponding digital information. Although this approach works fine in detecting the line, the analog information about the reflection rate of the floor which is related to whether or not the sensor is right on the line is lost. The datasheet in (Vishay Semiconductor, 2000) shows the reflective collector current of the circuit in figure 3 can give us the information how far the reflective optical sensor is from the line. Therefore, the output voltage would gradually change as the reflective optical sensor moves away from the line. This is the main idea used in (Chan, 2003) and the proposed DIY line following robot. The method that this paper used to estimate the line position will be addressed in the following section after the characteristics of the 7 reflective optical sensors are normalized. Although the reflective optical sensors used in the proposed line following robot are all CNY70, they may not possess the same characteristics. Therefore, the outputs of each reflective optical sensor circuit shown in figurer 3 should be normalized, such that the analog output values can later be used to find out the correct line position.

The analog output values of each reflective optical sensor circuit on the reference gray scale pattern are first collected. The values in figure 4 show that the 5 reflective optical sensors actually do not give identical outputs even the test conditions are the same. The normalization process tries to map the output values of each reflective optical sensor with respect to 20% and 80% of gray to the same maximum and minimum values by using the following formula:

$$y_{jo}=y_{\min}+\frac{y_{\max}-y_{\min}}{x_{\max,i}-x_{\min,i}}(x_{ji}-x_{\min,i}), \tag{2}$$

where $x_{max,i}$, $x_{min,i}$, are the maximum and minimum values of $i$th reflective optical sensor outputs, $y_{max}$, $y_{min}$, are the predefined maximum and minimum values for all reflective optical sensor outputs, $x_{ji}$ is the $j$th output value for the $i$th reflective optical sensor, and $y_{jo}$ is the normalized output value for $x_{ji}$.

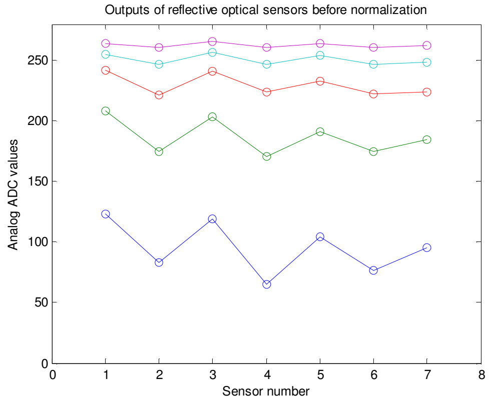


Fig.4 The reflective optical sensor outputs for different gray scale before normalization

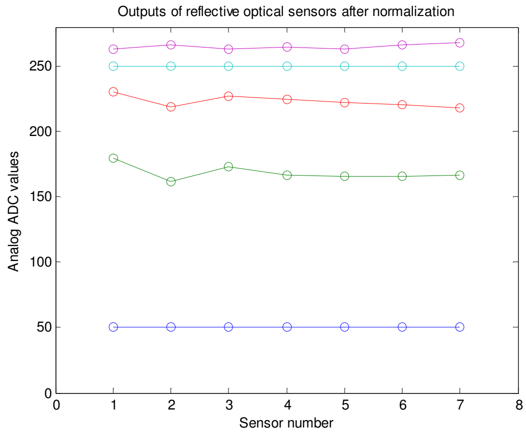


Fig.5 The reflective optical sensor outputs for different gray scale after normalization

It can be seen in figure 5 that the output values of different reflective optical sensors are quite similar to one another after the normalization process is done by using (2). Once the normalization process for the sensors is done, the output values of these 7 reflective optical sensors are ready to be used in estimating the line position.

### *2.3 The line detection algorithm*

Assume that the coordinate of the left most sensor is 0, and the distance between two consecutive sensors are considered as 1. The output of the reflective optical sensor circuit in figure 3 is higher when the sensor is closer to the line. Therefore, one can always find 3 consecutive sensors with higher output readings than the other 4 sensors. Assume that the coordinate of these 3 sensors are $x_1$, $x_1$+1, and $x_1$+2, and the true shape of the sensor output values in the range of [$x_1$, $x_1$+2] can be approximated by a quadratic curve. One can then find the following relationships between the coordinate of the sensors and the output values:

$$y_1 = ax_1^2 + bx_1 + c\,,\quad y_2 = a(x_1+1)^2 + b(x_1+1) + c \qquad (3.a)$$

$$y_3 = a(x_1+2)^2 + b(x_1+2) + c\,. \qquad (3.b)$$

The coordinate value at which the output value of the quadratic curve is the maximum is considered as the true position of the line. By using the basics of calculus, one would know that the coordinate value is

$$x = -\frac{b}{2a}\,. \qquad (4)$$

From equation (3-4), one can find that the true line position can be estimated by using the following formula:

$$a = \frac{y_1 + y_3 - 2y_2}{2}\,,\quad b = y_2 - y_1 - 2ax_1 - a\,. \qquad (5)$$

Figure 6 shows the comparisons of the true line positions and the estimations by using (3-5) with and without the proposed normalization process. It can be clearly seen in figure 6 that the normalization process plays a very important role in the line detection algorithm, and the estimation is quite satisfactory. This is also a practical implementation issue that students can learn in this laboratory.

### *2.4. The tracking controller simulation and implementation*

A proportional-derivative (PD) controller is used to drive the robot along the line. The performance of the controller was first simulated in the SIMULINK environment. The model file in figure 7 simulates the dynamics of the line following robot by using the formula in (1) whose inputs are the voltage commands for the dc gear motors of the right and left wheels, and outputs are the plenary coordinate and orientation of the line following robot. Low pass filters are used to simulate the responses of the dc motor speeds with respect to input voltages. The error between the center of the reflective optical sensors and the line to be followed is then processed by the PD controller to generate velocity commands for the right and left wheels. The proportional term is only a gain amplifier, and the derivative term is applied in order to improve the response to disturbance, and it also compensate phase lag at the controlled object. The D term will be required in most case to stabilize tracking motion. Although the integral control is not used, the steady state error is small enough and does not cause too much trouble.

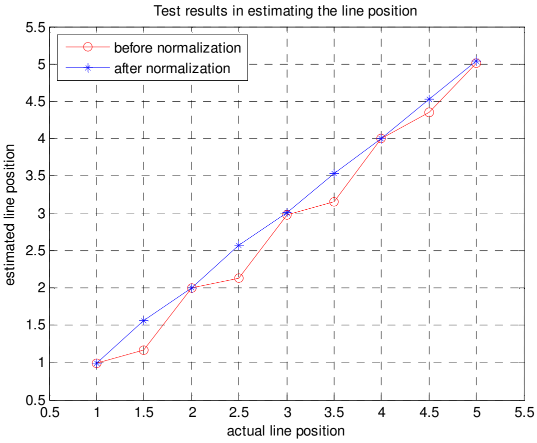


Fig.6 The comparisons of the true line position and the estimations with and without the normalization process

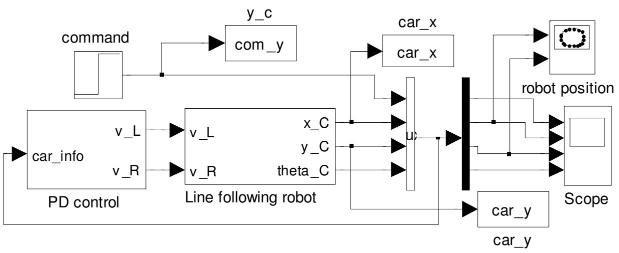


Fig.7 The SIMULINK model for the line following robot

The simulation results in figure 8 shows the performance of the PD controller. It can be seen that if the gain values of the PD controller is carefully adjusted, the line following robot can still follow a reference trajectory with an abrupt change of its path.

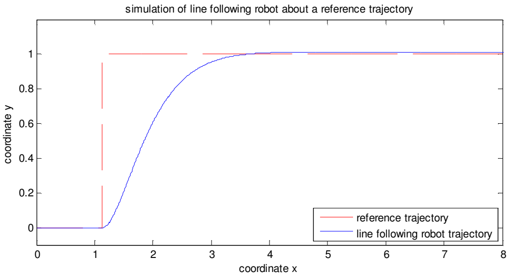


Fig.8 The simulation results of the line following robot using PD controller and the line to be followed

## 3. OVERVIEW OF THE MICROMOUSE

The micro-mouse devised in this section can be used to join local and international micro-mouse contests, though the kit is used as a learning aid for the hands-on laboratory. The students would learn at this part of the hands-on laboratory how to control the micro-mouse to go straight, to take 90 or 45 degree turns, to calculate the current position of the micro-mouse, to keep a safe distance away from the walls, to collect the maze information by using the reflective optical sensors, and to find the shortest path by using the popular Bellman flooding algorithm (Harrison, 2007, Watson, 2000). Because the software algorithms involved in this part of the hands-on laboratory, it is suggested that students should focus their effort on implementing the basic vehicle motion control, collection of the maze information, localization error correction by using the reflective optical sensors, and the shortest path searching algorithms.

### *3.1 Hardware*

The microcontroller used in the micro-mouse kit is quite similar to that of the line following robot. Because of the on chip RAM size, dsPIC6011 is chosen instead of the dsPIC4011. The three main algorithms in the firmware of the micro-mouse are the localization algorithm, the vehicle control algorithm, and the searching algorithm. The encoders of the dc motors and gyro are used mainly for the localization and vehicle control algorithms. Although the approximation errors may be accumulated in the localization algorithm, the optical sensors can be used to help correct them. Figure 9 shows the prototype implementations of the micro-mouse, respectively.

### *3.2 The searching algorithm and simulation program*

The popular Bellman flooding algorithm implemented in the micro-mouse is used to find the shortest path between any two cells in the maze by using the wall information stored in the RAM space. If the micro-mouse is at such a cell in the maze that all the surrounding cells it can move to were all visited before, the flooding algorithm will then help find a path to lead the micro-mouse to those unvisited cells. Such an idea is implemented as M-files in the MATLAB environment, and they are given to students such that they can study how the algorithm works and how to construct a shortest path according to partial information of the entire maze. To give students more experiences of how the searching and flooding algorithm work, a simulation program shown in figure 10 is also given to students. The algorithms also find the guiding information at each cell of the shortest path such that the micro-mouse can be moved from the start cell to the destination cell.

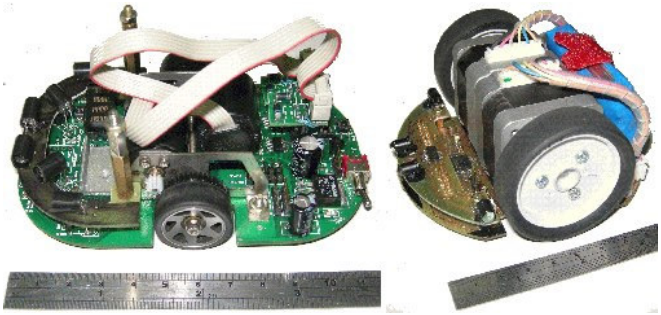
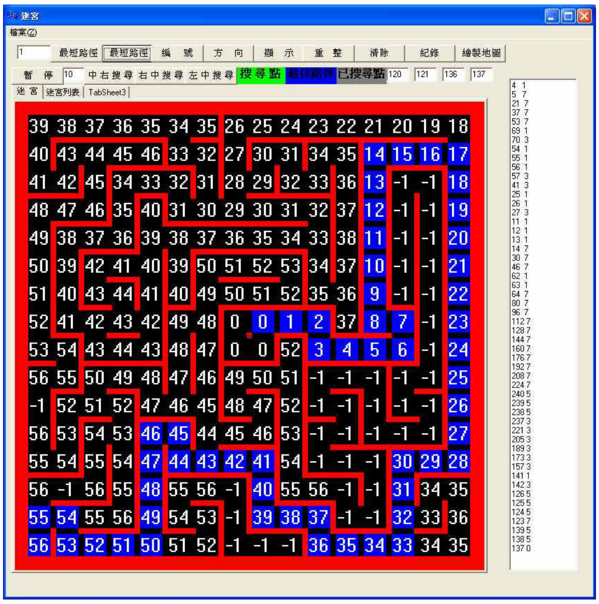


Fig.9 The prototype implementations of the micro-mouse



|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 39 | 38 | 37 | 36 | 35 | 34 | 35 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 |
| 40 | 43 | 44 | 45 | 46 | 33 | 32 | 27 | 30 | 31 | 34 | 35 | 14 | 15 | 16 | 17 |
| 41 | 42 | 45 | 34 | 33 | 32 | 31 | 28 | 29 | 32 | 33 | 36 | 13 | -1 | -1 | 18 |
| 48 | 47 | 46 | 35 | 40 | 31 | 30 | 29 | 30 | 31 | 32 | 37 | 12 | -1 | -1 | 19 |
| 49 | 38 | 37 | 36 | 39 | 38 | 37 | 36 | 35 | 34 | 33 | 38 | 11 | -1 | -1 | 20 |
| 50 | 39 | 42 | 41 | 40 | 39 | 50 | 51 | 52 | 53 | 34 | 37 | 10 | -1 | -1 | 21 |
| 51 | 40 | 43 | 44 | 41 | 40 | 49 | 50 | 51 | 52 | 35 | 36 | 9 | -1 | -1 | 22 |
| 52 | 41 | 42 | 43 | 42 | 49 | 48 | 0 | 0 | 1 | 2 | 37 | 8 | 7 | -1 | 23 |
| 53 | 54 | 43 | 44 | 43 | 48 | 47 | 0 | 0 | 52 | 3 | 4 | 5 | 6 | -1 | 24 |
| 56 | 55 | 50 | 49 | 48 | 47 | 46 | 49 | 50 | 51 | -1 | -1 | -1 | -1 | -1 | 25 |
| -1 | 52 | 51 | 52 | 47 | 46 | 45 | 48 | 47 | 52 | -1 | -1 | -1 | -1 | -1 | 26 |
| 56 | 53 | 54 | 53 | 46 | 45 | 44 | 45 | 46 | 53 | -1 | -1 | -1 | -1 | -1 | 27 |
| 55 | 54 | 55 | 54 | 47 | 44 | 43 | 42 | 41 | 54 | -1 | -1 | -1 | 30 | 29 | 28 |
| 56 | -1 | 56 | 55 | 48 | 55 | 56 | -1 | 40 | 55 | 56 | -1 | -1 | 31 | 34 | 35 |
| 55 | 54 | 55 | 56 | 49 | 54 | 53 | -1 | 39 | 38 | 37 | -1 | -1 | 32 | 33 | 36 |
| 56 | 53 | 52 | 51 | 50 | 51 | 52 | -1 | -1 | -1 | 36 | 35 | 34 | 33 | 34 | 35 |

|  |  |
|---|---|
| 4 | 1 |
| 5 | 7 |
| 21 | 7 |
| 37 | 7 |
| 53 | 7 |
| 69 | 1 |
| 70 | 3 |
| 54 | 1 |
| 55 | 1 |
| 56 | 1 |
| 57 | 3 |
| 41 | 3 |
| 25 | 1 |
| 26 | 1 |
| 27 | 3 |
| 11 | 1 |
| 12 | 1 |
| 13 | 1 |
| 14 | 7 |
| 30 | 7 |
| 46 | 7 |
| 62 | 1 |
| 63 | 1 |
| 64 | 7 |
| 80 | 7 |
| 96 | 7 |
| 112 | 7 |
| 128 | 7 |
| 144 | 7 |
| 160 | 7 |
| 176 | 7 |
| 192 | 7 |
| 208 | 7 |
| 224 | 7 |
| 240 | 5 |
| 239 | 5 |
| 238 | 5 |
| 237 | 3 |
| 221 | 3 |
| 205 | 3 |
| 189 | 3 |
| 173 | 3 |
| 157 | 3 |
| 141 | 1 |
| 142 | 3 |
| 126 | 5 |
| 125 | 5 |
| 124 | 5 |
| 123 | 7 |
| 139 | 5 |
| 138 | 5 |
| 137 | 0 |

Fig.10 A simulation program for the searching and flooding algorithms

### *3.3 Localization algorithm*

To estimate the orientation and position of the micro-mouse, the calculation of the linear velocity $v$ and rotational angular velocity $\omega$ from rotational speeds of the right and left wheels in (1) is used. The estimation is performed by using the following formulas:

$$\omega(n)=\frac{v_R(n)-v_L(n)}{2L},\ \ \theta(n)=\theta(n-1)+T_s\omega(n)\,, \tag{6.a}$$

$$v(n)=\frac{v_L(n)+v_R(n)}{2}\,, \tag{6.b}$$

$$x(n)=x(n-1)+T_s v(n)\cos(\theta(n))\,, \tag{6.c}$$

$$y(n) = y(n-1) + T_s v(n)\sin(\theta(n))\,, \tag{6.d}$$

where $v_L$ and $v_R$ stand for the linear velocities of the left and right wheels, $T_s$ is the sampling interval of the localization algorithm, and $\theta(n)$, $x(n)$, $y(n)$ are the orientation, and coordinate of the center point between the two wheels, respectively. The estimation by using (6.a-d), though not exact, is satisfactory if the sampling interval $T_s$ is chosen to be small enough. This is verified in the following paragraph which is related to the trajectory control of the micro-mouse.

### *3.4 The navigation algorithm and simulation*

Although the software algorithms which help the micro-mouse find the destination cell and the shortest path are introduced in the above section, the micro-mouse still has to move itself in the maze. The micro-mouse has to keep track of its localization information by using the encoders, the reflective optical sensors and the estimation formulas (6.a-d). The information and approaches provided by Peter Harrison (2007) are used to introduce to students how navigation of the micro-mouse is done in the maze. For example, velocity profiles are used for right and left dc motors to control the micro-mouse to go straight and to make smooth turns. A SIMULINK behavior model shown in figure 11 is used first to make students familiar with the algorithm. PD controllers are used for the speed controls of the left and right motors, and also the trajectory adjustment for the micro-mouse. The simulation results for the micromouse to adjust its y-axis position in the presence of noises are shown in figure 12. The SIMULINK behavior model is a good starting point for the students to know more about the micro-mouse dynamics, and how intuition and theories are applied in micro-mouse construction.

The orientation error correction algorithm can later be added into the SIMULINK behavior model, which can then be used to see how the localization algorithm in (6.a-d) should be corrected to compensate the orientation and position errors. This is the most time-consuming part of the micromouse hands-on laboratory.

## 4. CONTESTS AND FEEDBACKS

To make the hands-on laboratory more interesting to the students, contests for line following robots and micro-mouse will be held after the hands-on laboratory is finished. Well implemented line following robots and micro-mouses can also join national (Fortune Institute of Technology, 2007, Southern Taiwan University, 2007) and international contests such that the students can also learn from the other competitors during the contests. The students' feedbacks about this hands-on laboratory are collected by using questionnaires. It can be seen from table 2 that the feedbacks of the questionnaires from the students are quite positive.

## 5. CONCLUSIONS

A hands-on laboratory for autonomous robot design courses is presented in this paper. The hands-on laboratory is divided into two parts. The first part helps students learn the basics of hardware circuits and software control algorithms for autonomous robots by using a line following robot kit devised in the Lunghwa University of Science and Technology. The implementation issues in this part of the hands-on laboratory include vehicle control, collection and normalization of the sensor information, the local line detection algorithm, and the automatic line following control algorithm. The second part of the hands-on laboratory is a micro-mouse which contains more complex vehicle control, localization, and searching algorithms. The students would learn more advanced programming and implementation skills at the second part of the hands-on laboratory. Implementation kits of the robots are given to students to lessen the learning burden, and help them easily acquire the knowledge in design and implementations of autonomous mobile robots. The feedbacks of the students are quite positive, and the hands-on laboratory is also supported free of charge the necessary microcontrollers.

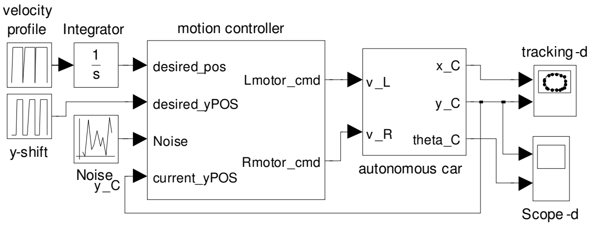


Fig.11 A SIMULINK behavior model for a micro-mouse and the trajectory following algorithm

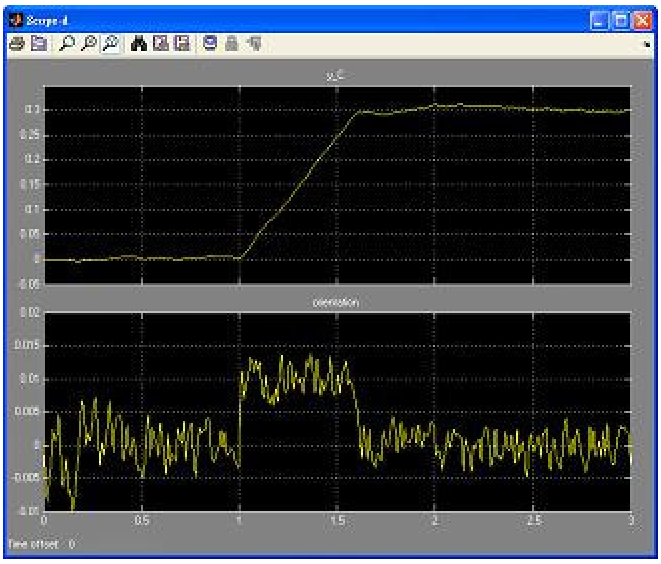


Fig.12 The simulation results for the micromouse to adjust its y-axis position to 0.3 in the presence of noises

Table 2. The average scores of the hands-on laboratory questionnaires

| **Questions** | **Average score (1-5)** |
|---|---|
| The hands-on laboratory can effectively help me learn hardware circuit and programming implementation skills | **4.06** |

| Questions | Average score (1-5) |
|---|---|
| The contests held after the mobile robots are finished interest me a lot and encourage me to learn more about the necessary skills and theories. | **3.91** |
| It is more interesting to me to make the robot from scratch in both hardware and software, because each part are explained in details. | **4.15** |
| I am willing to pay the money to own the mobile robot. | **3.52** |

## REFERENCES

Chan, "Desktop line following robot," http://elm-chan.org /works/ltc/report.html, Oct. 2003.

Fortune Institute of Technology, The line following robot speed contest 2007, http://www.ee.fjtc.edu.tw/racepage /index.htm.

Harrison, P., "Micromouse Information Center", http://micromouse.cannock.ac.uk/, 2007.

IEEE CSUC student chapter, Micromouse, http://www.csuchico.edu/ieee/micromouse.html, 2004.

iRobot cooperation, Vacuum cleaning robot - iRobot Roomba, www.irobot.com, 2007.

Linda Geppert, "QRIO – The robot that could," IEEE Spectrum, pp. 34-37, May, 2004.

Microchip Technology, Inc., "dsPIC30F4011/4012 datasheet," 2007.

National Aeronautics and Space Administration (NASA), the robotics alliance project, http://www.robotics.nasa.gov/, 2007.

Qu, Z., and X. Wu, "A new curriculum on planning and cooperative control of autonomous mobile robots," *International Journal of Engineering Education*, Vol. 22, No. 4, pp. 804-814, 2006.

Shayang YE Industrial Co., Ltd. Micro DC Planetary Geared Motors, http://www.shayye.com.tw/chinese/index-c.htm, 2000.

Southern Taiwan University, 2007' Taiwan Micro Mouse Contest, http://www2.eecs.stut.edu.tw/~mouse2007.

Thurmond, A., Seymour Papert & Constructionism, http://online.sfsu.edu/~foreman/itec800/finalprojects/annmariethurmond/home.html, 1999.

Vroman, J., "JavaBot1…. A line following robot," http:// www.james.vroman.com/javbot1a.htm.

Vishay Semiconductor GmbH, "CNY70 – Reflective optical sensor with transistor output," http://www.vishay.com/docs/83751/ cny70.pdf, Rev. A4, April, 2000.

Vishay Semiconductor GmbH, "CNY70 – Reflective optical sensor with transistor output," http://www.vishay.com/docs/83751/cny70.pdf, Rev. A4, April, 2000.

Watson, G., Micromouse motor controller and maze solver, May 2000.