The application of micromouse and MATLAB in teaching autonomous mobile robots

Juing-Huei Su, Chyi-Shyong Lee, and Hsin-Hsiung Huang
Department of Electronic Engineering
Lunghwa University of Science and Technology
No. 300, Sec. 1, Wan-Shou Rd., Touyuan county
Taiwan
suhu@mail.lhu.edu.tw

Abstract: - The application of micromouse and MATLAB to raise student interest in learning the implementation skills of autonomous mobile robots is presented in this paper. The project-oriented hands-on laboratory is devised for graduate students from the department of electronic engineering of Lunghwa University of Science and Technology, and for introductory workshops for vocational high school students and teachers with electronic and information engineering backgrounds. To enhance the learning outcomes, micromouse contests are also organized for students to see how well they learned the skills in the laboratory. The assessments and feedback from students show that the kit and the related laboratory experiences do help motivate students to learn actively and successfully the implementation skills of autonomous mobile robots.

Key-Words: - MATLAB, micromouse, mobile robot

1 Introduction

Robots are attracting more and more people's attention recently [1], especially when iRobot announced the Roomba vacuum cleaning robot [2]. This makes the fundamental understanding of robots a necessity for many electronic engineers [3]. Unfortunately, learning the design of robots is interesting but difficult, because it includes many areas of knowledge, e.g., mechanics and electronics, automatic control, and firmware programming of microcontrollers, etc. It is found that students will be willing to do tedious research work to solve practical problems when these problems are related to an interesting, and competitive contest [4]. The results also showed the micromouse when integrated with conventional curriculum is very helpful in motivating students and keeping their interest high. This is an important factor for technology oriented university students in Taiwan, because of their low learning achievements in traditional lecture oriented courses. Although the original micromouse contest in which autonomous robots compete for the title of best speed and intelligence was started about 30 years ago, these contests are still very popular to engineering students in UK, USA, Japan, Singapore, and Taiwan [5-9], because the micromouse is still evolving to perform better. Those state of the art micromice can even run at a speed of 3m/sec and in a diagonal path.

One of the problems with commercially available mobile robot kits [10-12] for autonomous mobile robot courses is that they are usually expensive. Therefore, a low-cost micromouse kit is devised in the Lunghwa University of Science and Technology to help not only raise student interest, but also motivate them to learn actively the implementation skills and related theories about autonomous mobile robots. The kit has been used in free workshops for 37 vocational high school students and 24 teachers with electronic and information engineering backgrounds in 2009, 2010, and in a mobile robot course for 9 graduate students from the department of electronic engineering in Lunghwa University of Science and Technology, in 2010.

A project-based laboratory based on the kit and MATLAB/SIMULINK is also developed to lead students step by step in making the micromouse search successfully the goal square of the maze. The contents includes 1) introduction of the integrated development environment (IDE), 2) interrupt driven real-time control firmware, 3) a simple maze-solving algorithm, 4) maze wall detection via infrared light emitting diodes (LEDs) and optical sensors, 5) motion control of stepping motors and the micromouse, and 6) adjustments of motion commands for position and orientation errors of micromouse. The last part of the course is either a micromouse contest or peer evaluating oral presentations for students to see how well the students have learned from the laboratory.

2 The micromouse kit

The kit shown in Fig. 1 is steered with step motors and differential drive. The firmware can be downloaded to the flash memory of the microcontroller dsPIC30F6010 through an in system programming (ISP) port. Students can collect the data stored in the microcontroller, or send commands to the firmware via an RS232 serial bus. The micromouse controls 6 infrared light emitting diodes (LEDs) in 5 directions, and detects the intensity of the reflected light to determine the maze wall information and to correct the motion commands for two step motors. The kit can also keep track of its position in the maze by using the pulses which are sent to step motors. The firmware in the microcontroller can interact with the user with buttons and matrix LED display. By using the functions the mircromouse kit provides, students should devise their own maze-solving algorithm to help the micromouse find out the goal and decide an optimal route from the start to the goal according to its motion capability.

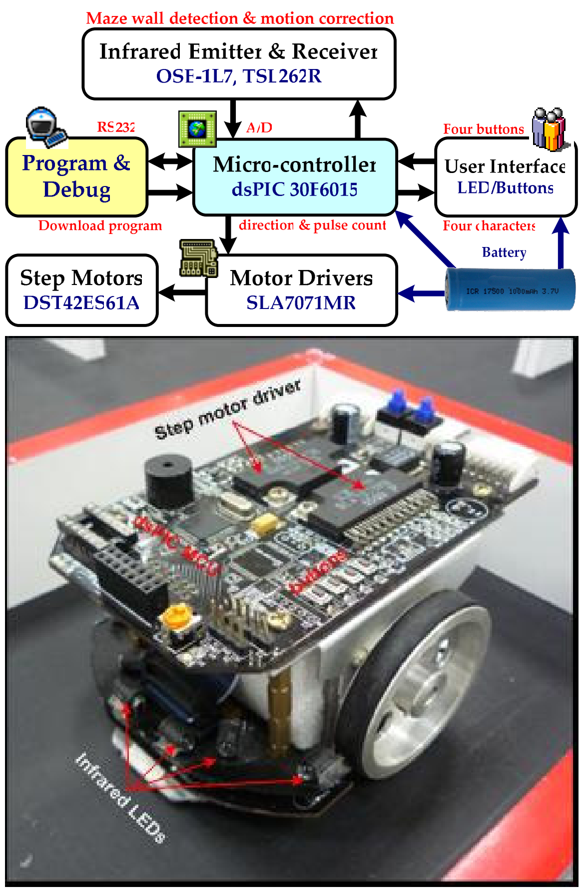

Fig. 1 The low cost micromouse kit designed for teaching autonomous mobile robots

3 The hands-on laboratory

The project-based hands-on laboratory exercises are described in details as follows.

3.1 An interrupt driven real-time control firmware

The micromouse has to calculate its position in a maze and scan the environment with the infrared LEDs and optical sensors constantly, such that the errors in position calculations can be corrected and the goal cell can be found. This relies heavily on an interrupt driven real-time control firmware structure. Students would learn in this part how to control the time intervals of system functions for peripherals, motion control, and the maze solving algorithm. These time-intervals should be short to make the micromouse react fast enough to environment changes, which is important when the micromoue is running fast. Key factors to influence the code execution efficiency, such as fixed-point mathematics, table-lookup skills for trigonometric functions, and programming skills would be introduced to students in this part.

3.2 A simple maze-solving algorithm

There are many ways to solve the mazes found in the micromouse competitions. The simplest method is some variation on the flood-fill or Bellman algorithm [13-14]. The idea is to start at the goal square and fill the maze with values which represent the distance from each square in the maze to the goal square. When the flood reaches the start square, the algorithm can then be stopped. The micromouse could follow the values downhill to the goal square. Although the micromouse knows nothing but the start square about the maze configuration at first, it can still follow a route suggested by the flood algorithm if unvisited squares are assumed to contain no maze walls. When entering an unvisited square, it records the wall information and remembers the visited status of that square. By using the procedure [13] described above, the goal square would be found at last.

When the goal square and the shortest paths from the start square to the goal square are found, the shortest paths can be furthermore optimized such that the micromouse can run in diagonal instead of consecutive 90 degree turns to reduce even more the run time from the start square to the goal square. The optimization process can also be used to select a best route for the micromouse among the routes suggested by previous flood-fill algorithm. Fig. 2 shows a MATLAB program for students to learn those basic concepts of a simple maze-solving algorithm.

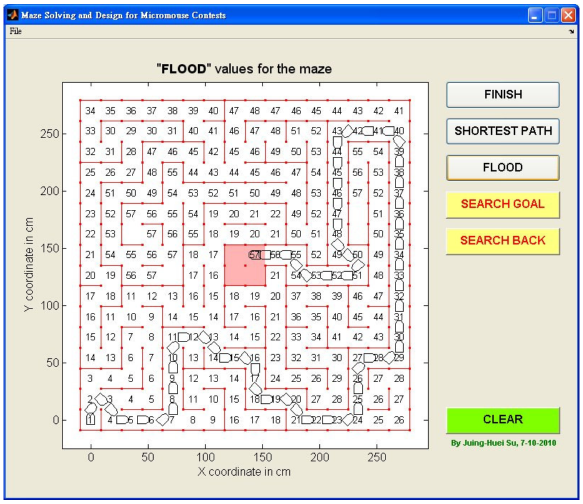

Fig. 2 A MATLAB program for learning maze solving algorithms

3.3 Maze wall detection via infrared LEDs and optical sensors

The micromouse kit uses infrared LED and optical sensor TSL262R to collect wall information when moving in a maze. According to the datasheet of TSL262R, the irradiance (W/cm^2) is proportional to its output voltage in log scale. After some mathematical manipulations, the relationship of the optical sensor output voltage v_O and the distance r between the LED and the wall can be described as

$$\log v_O = \alpha \log r + \beta , \quad (1)$$

where are unknown constants to be determined. The coefficients in (1) can be calculated by using experimental data, and the following formula from least square methods.

$$\begin{bmatrix} \log r_1 & 1 \\ \vdots & \vdots \\ \log r_m & 1 \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \log v_{O1} \\ \vdots \\ \log v_{Om} \end{bmatrix} \Rightarrow \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = (A^T A)^{-1} A^T B . \quad (2)$$

The result in Fig. 3 clearly shows the experimental data and theoretical prediction by using (3-4) agree with each other very well.

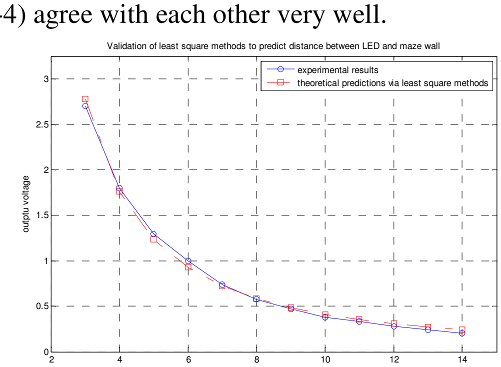

Fig. 3 Comparisons of the experimental data and theoretical predictions by using (1)

The approach is a way for the micromouse to find out the distance to the maze wall, and therefore determine the maze wall configuration in a single maze cell.

3.4 Control of step motors and micromouse

The micromouse kit uses step motors to move itself around in a maze. Therefore, the skill to control the acceleration, and speed of step motors is the most fundamental part in using the kit. The step motor driver used in the kit controls the step motor according to input pulses. The time interval between these consecutive input pulses determines the speed of step motors. Fig. 4 is an example of how to control the speed and acceleration of step motors. Since the diameter of the wheel is 51mm, and the default running mode for the step motors is 400 pulses/round, the velocity and acceleration of the step motors can be calculated by using the following formulas:

$$v_i = \frac{\pi \times 51}{400 \times T_i} \text{(mm/sec)}, \; i = 1, 2, 3, \ldots, \quad (3)$$

$$a_i \approx \frac{v_{i+1} - v_i}{T_i} \text{(mm/sec}^2) \Rightarrow T_{i+1} \approx \frac{51\pi \times T_i}{400 a_i \times T_i^2 + 1} . \quad (4)$$

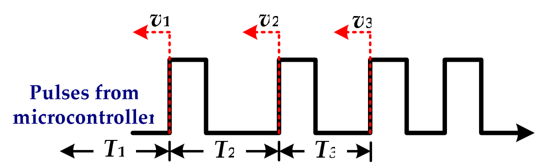

Fig. 4 Speed and acceleration calculations for step motors controlled by pulses

Equation (4) is used to determine how large the time interval T_{i+1} should be when given T_i and the acceleration a_i. In practice, the acceleration a_i should be changed gradually and limited by the output torque of the step motors to prevent loss of steps. Fig. 5 shows how the speed and acceleration profiles are planned according to equation (4) and an acceleration limit of 5m/sec^2 for the step motor. These profiles would be stored in the firmware of the micromouse kit to save time for the microcontroller.

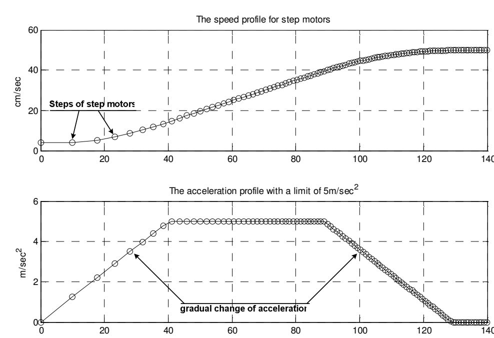

Fig. 5 Speed and acceleration profiles for step motors by using (4)

The second part of this section is to introduce to students how pulse commands of step motors can be used to calculate the posture of the micromouse. Assume that the orientation at first of the micromouse is . as shown in Fig. 6b, and the distance between two wheels is *L*. It can be seen from Fig. 6 that the following relationships in Table 1 hold.

Table 1 Position and orientation calculations for step motor micromouse

Right wheel moves a step
orientation: $\theta_n = \theta + \Delta\theta$
position
$x_{Cn} = \frac{1}{2}(x_{Rn} + x_L)$, $y_{Cn} = \frac{1}{2}(y_{Rn} + y_L)$ $\begin{bmatrix} x_{Rn} - x_L \\ y_{Rn} - y_L \end{bmatrix} = \begin{bmatrix} \cos\Delta\theta & -\sin\Delta\theta \\ \sin\Delta\theta & \cos\Delta\theta \end{bmatrix} \begin{bmatrix} x_R - x_L \\ y_R - y_L \end{bmatrix}$

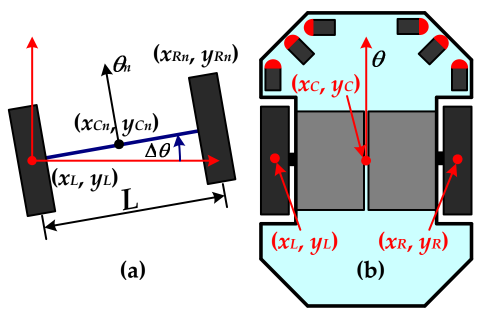

Fig. 6 Position calculation of the micromouse when right or left step motor moves one step

Unfortunately, if the calculations in Table 1 use float numbers, the execution time would be too long for the micromouse to react to environment changes when running faster than 1.5m/sec. Therefore, students would have to learn at this part the fixed point arithmetics in order to make their firmware code more efficient.

3.5 Motion control with error corrections

It is interesting to know even pulse commands with the same frequency can not guarantee that the micromouse will go in a straight line. This is because the phase differences of pulse commands for the two step motors, and the load conditions for the two wheels are eventually not the same. Therefore, optical sensors discussed in previous section to measure distance between micromouse and maze walls should be used to correct longitudinal, lateral, and alignment errors shown in Fig. 7. The alignment and lateral errors of the micromouse can be corrected by using the infrared sensor readings and the pulse commands for step motors. The longitudinal error can be corrected if the pole position of the maze can be detected when the micromouse runs in a given maze. Because the initial position and the orientation angle of the micromouse are set to 0 and /2, respectively, these calculation errors should also be corrected based on the infrared sensor readings. Diagonal straight line motion can be corrected by using the readings of forward-looking optical sensors, such that the micromouse would not hit the posts or corners alongside the path.

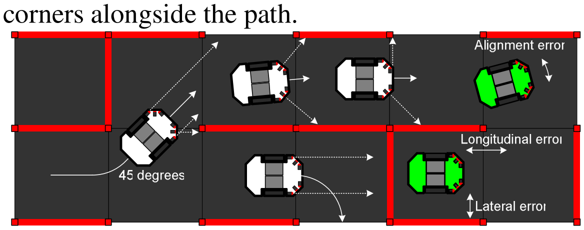

Fig. 7 The ways that a micromouse uses in correcting the longitudinal, lateral, and alignment errors

To make the firmware work efficiently, the velocity profiles which generate smooth trajectories for step motors would be calculated according to the diameter of wheels, the acceleration limit, the time to make desired turns, and the micromouse position when leaving the turn movement and starting the straight line movement. These velocity profiles are very important because the turns for the micromouse in the maze are mostly blind [15]. There is not much information which can be used to correct the micromouse movements in making turns. Therefore, simulation of the micromouse trajectory is necessary to save time and to determine appropriate speed and acceleration profiles. The simulation results shown in Fig. 8 for a micromouse to make 90 degree turns with a radius of 8cm can be obtained by using sample MATLAB m-files which are given to students in class. Students can try different average speed and turn radius by themselves, and implement the results into their own firmwares.

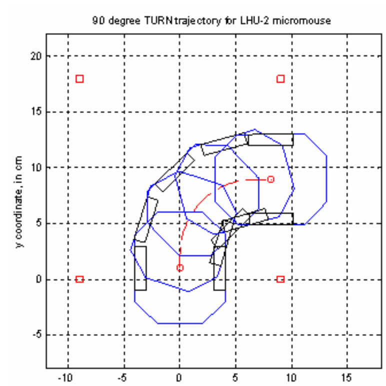

Fig. 8 Simulation results for a micromouse to make 90 degree turns

4 Workshops, Contests and feedbacks

The micromouse kit described in this paper is used in 2009 to encourage vocational high school teachers and students to learn implementation skills of intelligent mobile robots. Figure 9 shows the free workshops supported by the Ministry of Education. Moreover, every participant of the workshop can bring back not only the micromouse, but also the corresponding IDEs. It can be seen from Table 2 that the feedback on the survey questionnaires from vocational high school students (22 in 2009, 15 in 2010) and teachers (18 in 2009, 6 in 2010) was quite positive in consecutive 2 years. The majority of the participants (86.6%, 92.4%) agreed that they were motivated to learn those skills and theories, and were satisfied (88.6%, 92.4%) with the organization of the laboratory. To see how well the participants of the workshop integrate all the skills learned in the hands on laboratory, contests (Figure 9) were held after they finished their micromice. It is observed that most of the participants do try actively to make their micromice run smoothly and win the contest.

Fig. 9 The micromouse kit used in a workshop and a contest for students' final projects

Table 2. The average scores of the survey questionnaires for the workshop

Questions (score: 1-5)	Mean (standard deviation)	
	2009	2010
The arrangement and design of the hands-on laboratory for micromouse is good	**4.45 (0.55)**	**4.62 (0.50)**
The handouts for the hands-on laboratory are effective and complete	**4.43 (0.68)**	**4.38 (0.67)**
The laboratory is helpful for students in improving hardware and software implementation skills, and in raise our interest	**4.33 (0.66)**	**4.62 (0.50)**
I am happy to join the workshop	**4.45 (0.68)**	**4.62 (0.50)**
I would recommend to my colleagues or classmates to join this workshop next time	**4.56 (0.53)**	**4.57 (0.51)**

5 Conclusion

A low cost micromouse kit developed in Lunghwa University of Science and Technology and used as a learning platform for autonomous mobile robots is presented in this paper. Several laboratory exercises with graphical user interface of MATLAB are also devised for the kit to help students learn the basics about micromouse and mobile robots. They include 1) introduction of the integrated development environment (IDE) for the micromouse kit, 2) interrupt driven real-time control firmware structure, 3) implementation of a simple maze-solving algorithm, 4) maze wall detection via infrared light emitting diodes (LEDs) and optical sensors, 5) motion control of stepping motors and the micromouse, and 6) adjustments of motion commands of stepping motors for position and orientation errors of micromouse in a maze by using infrared LEDs and optical sensors. The kit is designed with help from the local branch of Microchip Inc., therefore, students can get the necessary C-compilers and microcontrollers free of charge. It is hoped that every student can have his/her own micromouse easily even if he/she is economically disadvantaged, and learn quickly the techniques in making an autonomous mobile robot. The assessments and feedback from students show that the micromouse kit and related national contests do effectively raise students' interest, and therefore motivate them to learn actively those related theory and implementation skills. The overall cost of the kit is less than USD$150, which is well below the cost of similar commercially available kits [10, 11].

6. Acknowledgements
The authors would like to thank the National Science Council for its financial support under grant NSC 99-2511-S-262-001, and Microchip Inc. for their support of free C-compilers and microcontrollers.

References:

[1]. Bill Gates, "A robot in every home," *Scientific American Magazine*, Jan. 2007.
[2]. iRobot cooperation, Vacuum cleaning robot - iRobot Roomba, *www.irobot.com*, Accessed June 21, 2009.
[3]. 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.
[4]. Ning Chen, H. Chung, and Y. K. Kwon, "Integration of micromouse project with undergraduate curriculum: a large-scale student participation approach," *IEEE Trans. on Educ.,* Vol. 38, No. 2, pp. 136-144, May, 1995.
[5]. The UK micromouse contests website: http://www.tic.ac.uk/micromouse/
[6]. The 24th annual micromouse contest in the IEEE Appllied Power Electronics Conference and Exposition 2010, www.apec-conf.org.
[7]. The 2010 all Japan micromouse contest, http://www.ntf.or.jp/mouse, Accessed August 11, 2010.
[8]. The Singapore robotic games 2010 website: http://guppy.mpe.nus.edu.sg/srg/
[9]. The 2009 Taiwan micromouse & intelligent robot contest, http://robot2009.lhu.edu.tw
[10].Micromouse AIRAT2 website: www.robotstorehk.com/micromouse/RS-AIRAT2.html.
[11].Intelligent 8051 micomouse from ICtec, website: www.icci.com.tw/web/MdFront?mdId=MD0000002307000830&command=displayDetail
[12].LEGO mindstorm website: http://mindstorms.lego.com/en-us/default.aspx
[13].Society of Robots, "Micromouse – Maze Solver – Theory," website: http://www.societyofrobots.com/member_tutorials/node/94
[14].Peter Harrison, Micromouse Online, website: http://www.micromouseonline.com/blog/.
[15].Ng Beng Kiat's Homepage, http://www.np.edu.sg/alpha /nbk/, accessed on 3/18/2010.