Creating a Simulation Environment for the Micromouse

Martin Komák, Elena Pivarčiová

Technical university in Zvolen, Faculty of Technology, Študentská 26, 960 01 Zvolen, Slovakia

Abstract **– The emphasis on the autonomy of all production facilities is increasing. It is normal that the company's goal is to constantly increase its profits and thanks to automation to win the competition. In order to verify real solutions, simulation models could often be created before a real project. This saves time and investment in development. Deployment of a robotic mouse (micromouse) in production or other processes is becoming an increasingly common step. This article describes one of the ways to create a simulation model of a micromouse that will move around the labyrinth.**

The task of the robot is to reach the target position in the uncharted labyrinth. The goal is for the robot to avoids all obstacles on its way from the starting position to the end. The current program needs human intervention in control. However, we can say that this environment serves as a "cornerstone" for the further development of an autonomous control system. So far, we are considering the simple complexity of the labyrinth.

Keywords **– micromouse, labyrinth, collision detection.**

1. Introduction

Continual development in the field of automated systems provides us with increasing ergonomics in the workplace and facilitates human work. Thus, daily routine work often becomes a thing of the past. Tasks such as storage, selection and loading of goods, automatic control and many others are beginning to be performed using the so-called autonomous mobile robots (AMR). The practical use of this technology is also in the environment with hazardous substances or other environments with a threatening effect on life. Perhaps the most common use is material handling for transportation [1], [2]. Due to the fact that mobile robots move in a dynamic environment, their control system has to be at a high level. The research of autonomous mobile robots can be too high, so the main "burden of research" can be replaced by the development of so-called micromouse, which will represent a simplified electromechanical model of AMR.

Several authors have addressed the issue of micromouse and minirobots:

The literature [3] describes the states of how a robot should move optimally, accurately and safely in every situation. It is necessary that every obstacle placed in the transition trajectory is bypassed so that its robotic mouse avoids it. The optimal trajectory means choosing exactly the shortest and most appropriate specific movement to be completed.

Robots are necessary for increasing the production system productivity too [4]. The AMR area is classified as Industry 4.0, which is a popular topic of the product ecosystem today. Many businesses that have high profitability, sustainable product life and high efficiency will be able to integrate Industry 4.0. Industry 4.0 is a very broad area covering manufacturing processes, efficiency, data management, consumer relations, competitiveness and much more. Based on the obtained results, the archetypes of Industry 4.0 are described. The results obtained by the analysis of the literature not only summarize the current knowledge about Smart Factories, but also indicate the directions of potential research [5].

Article [6] presents a survey of innovative methods of driving electric vehicles. The basic characteristics of nonlinear system linearization, predictive control model and neural network are proposed.

From a control point of view, high-quality motion control requires precise position and speed signals.

DOI: 10.18421/TEM111-61
https://doi.org/10.18421/TEM111-61

Corresponding author: Elena Pivarčiová,
Technical university in Zvolen, Faculty of Technology, Študentská 26, 960 01 Zvolen, Slovakia.
Email: pivarciova@tuzvo.sk

Received: 21 December 2021.
Revised: 15 February 2022.
Accepted: 22 February 2022.
Published: 28 February 2022.

However, velocity estimation based on a simple numerical distinction from position measurement can be very erroneous, especially if the sampling frequency is high or the velocity is very low [7].

Authors in [8] study a new method of infrared sensor measurement for micromouse control.

Using robotics and automation especially for two-wheeled mini robots describes authors in [9] and applications of drives and converters in [10].

The IEEE Micromouse is a university-level competition in which student teams build small robots capable of researching mapping and racing in a small labyrinth. Building a successful micromouse involves hardware design and construction, control, sensor and system integration, microcontroller integration, and the development of computer algorithms for mapping and route planning [11], [12], [13].

In order for the micromouse to run in an unknown labyrinth, students have to integrate mechanical design skills, printed circuit boards, sensor calibration procedures, digital motion control algorithms, route planning algorithms, and create a labyrinth solver [14], [15].

Article [16] deals with methods for accelerating the operation of convolutional neural networks for autonomous robotic learning. An analysis of the theoretical possibilities of modifying the learning mechanism of the neural network is performed.

In our paper, we created a labyrinth of different path complexities as an environment model to simulate different nooks and ways. Such an environment will ensure that the micromouse model tends to adapt to any environment. It could be said that such an application represents a model of free movement with obstacles. Such a task can be solved in various ways. In our case, it is possible to look at it as a 2D task, looking at the labyrinth from above. Nevertheless, we chose all algorithms as if it were a 3D task due to higher variability and usability in the future. With autonomous control, the vehicle should perform work without human intervention throughout its operating time. The program created by us does not meet this yet, but it is created as a "background" for further development of control systems based on neural networks or other control algorithms. The results achieved by us focus on instructing how it is possible to create a simulation environment for micromouse.

2. Problem Identification

The issues addressed by this article could be summarized in two areas. The first would be how to create a visualization environment for microm.

The easiest way to create the environment in which the robotic mouse wiouse control design in any programming language. The second problem consists of moving the mouse around the environment, its control and solving the detection of collisions with obstacles. The conclusion is a simulation verification of all our procedures.

3. Creating an Environment (labyrinth)

ll move is to use CAD or another graphics tool. The labyrinth model is then simply read via a stl reader.

We "modelled" the labyrinth that we designed (Figure 1) directly in our program using the exact coordinates of the contours of the labyrinth and its partitions. All the vertical parts of the labyrinth are obstacles for the mouse. In the labyrinth, the start position is defined by the green color and the end position of the micromouse is defined by the red color of the floor.

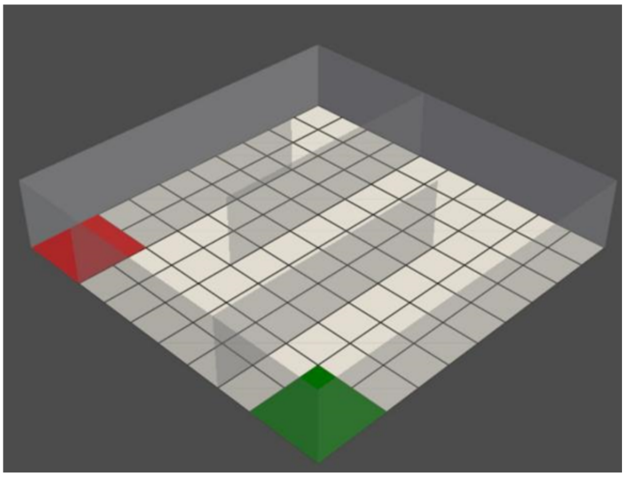

Figure 1. Created Labyrinth

The raised parts of the labyrinth are 4 contours and 3 partitions, which are placed so as to create a winding path for the mouse (see Figure 1). These parts are defined as areas with precise bounding vertices in the Cartesian coordinate system. This shape of the labyrinth was designed so that the robot could try rectilinear movement in all directions and turning corners with larger and smaller radius. The labyrinth can be freely changed, one just need to change its coordinates of the partitions, or add or remove them.

4. Micromouse Movement

To ensure that the mouse moves across the labyrinth, we have created methods that rotate the mouse and move the mouse step which have been selected in the direction of mouse rotation so that the mouse visually moves through space smoothly with sufficient speed. We always have to make sure that when one moves the mouse, its axis of rotation also moves.

5. Micromouse Control

We can move the robotic mouse in different ways. Among the basic methods of motion determination is control (in our case it will be arrows on the keyboard), a pre-programmed path, or more advanced algorithms, such as neural networks or fuzzy systems. The movement we choose is the simplest from the programming point of view, but at the same time the least autonomous. The whole control cycle ends if the mouse crashes into an obstacle (Figure 2a) or reaches the target (Figure 2b).

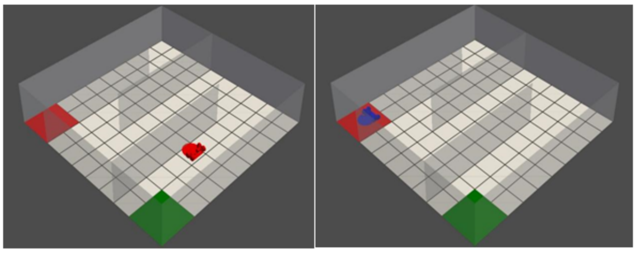

Figure 2. Micromouse in collision and in the target

Mouse control is in an endless while cycle asking if one of the arrows is pressed. When pressed, the function is automatically called, which moves the micromouse by the specified step (in our case by step 7 pixels). This step was chosen experimentally. When one presses the left or right arrow, the function of the mouse around its own axis of rotation is also programmed in the function.

6. Collision Detection

Identification of the collision of two objects (in our case part of our labyrinth and the micromouse) is necessary in this project. Imagine a real physical mouse with fine electronics and a modelled labyrinth made of some solid material. If we did not detect a collision between space and the mouse in time, it would quickly happen that a dynamic mouse crashes into an object, whether it is part of a labyrinth or another object located in the mouse's operating space. In order to simulate collisions, we create a function, the return value of which determines whether the mouse is in collision with any part of labyrinth or not. The input to the function is two objects that we will evaluate. One of the objects is always the mouse, as it is our only moving object and the other will be part of the labyrinth, whether it is a partition or the outer wall of the labyrinth. At first, we will read the vertices of the individual triangles that represent the surface of the objects in the cycle. This is a work with the so-called stl format. We will do the same for the second object. In order to be able to determine the penetration of one triangle into another, we create from one of them the general equation of the plane in which the given triangle lies. From the three vertices we define the parameters of the lines:

p1_objA = V2_objA - V1_objA
p2_objA = V3_objA - V1_objA

where V1_objA, V2_objA, V3_objA are vertices from object A and p1_objA, p2_objA are line parameters.

By the vector product of the parameters of the lines p1_objA and p2_objA we obtain the normal of the plane in which the triangle from object A lies. Substituting all the obtained values into the general equation of the plane we also obtain the value d. We create a general equation of the plane in which we substitute the vertices of object B. This way we get 3 solutions. To obtain the intersection between the 2 triangles, the following must apply:

(sol1 > 0 and sol2 < 0 and sol3 < 0) or (sol1 < 0 and sol2 > 0 and sol3 < 0) or
(sol1 < 0 and sol2 < 0 and sol3 > 0) or (sol1 < 0 and sol2 > 0 and sol3 > 0) or
(sol1 > 0 and sol2 < 0 and sol3 > 0) or (sol1 > 0 and sol2 > 0 and sol3 < 0)

where sol1 to sol3 are solutions of the general equation of the plane.

This is how we get the result if the plane intersects the triangle. Since we have to compare two triangles (which is just a bounded element from the plane), we have to verify these relations in reverse by calculating the general equation of the plane from the triangle of object B and then substituting the vertices of the triangles from object A. This not only calculates the intersections plane – triangle but also triangle - triangle.

7. Creating the Cover for the Micromouse

Due to the higher computational power and thus the lower computation time, it is possible to choose the collision identification only by using the so-called cover that will represent the object. The cover (Figure 3) can be created around an object as a shell that will simply represent the object. A cover is the creation of invisible planar walls around an object that will look like a cube or cube without top and bottom walls (omitting these unnecessary walls will increase our computing power). We do not need these two walls because the robotic mouse will only move in the x and y coordinates and the height remains unchanged, the robot will move constantly on the floor. The labyrinth has no ceiling. We create a cover by creating the vertices of a triangular network representing the walls from the limit points of the given object. We find the maximum and minimum value in all axes of the coordinate system between the vertices of the triangles representing the area of the object. These vertices will determine where the cover is currently located. We cannot forget that we will move the whole cover and rotate it around the space together with the robot. Otherwise, we would not capture the current position of the robot and we would record possible collisions elsewhere. Therefore, we add the movement of the entire cover to the program where we move with the robot. The cover will rotate around the robot's axis of rotation in a vertical axis parallel to the global coordinate axis z. We can still modify the given envelope by moving the limit coordinates of the peaks away from the object, in order to prevent real collisions arising from the inaccuracy of the production of a real mouse.

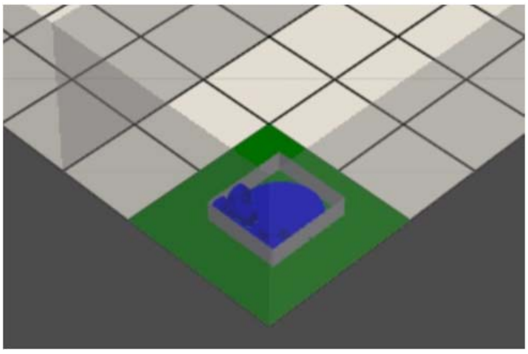

Figure 3. Illustration of the cover around the micromouse

We do not display the mouse cover in the resulting program to ensure more beautiful program graphics. The micromouse was designed in a CAD CAM program.

8. Simulation

We verify all programmed functions, methods and commands by the simulation. It is necessary to verify their logical functionality and overall conceptual solution. The goal is for the mouse to be able to move with the help of the arrows in the free space without collisions and after touching the obstacle it lights up red and the program stops (Figure 4). To control the mouse around space, we used an algorithm in which we compress the individual directions of movement of the micromouse using arrows (Figure 4).

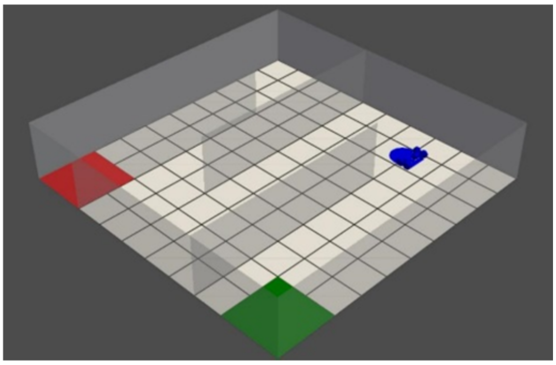

Figure 4. Micromouse in motion

9. Discussion

Another possibility is that we do not take all graphic parts of the mouse into the collision identification, but simplify its model. In practice, this means that we can render the mouse in a simplified way and not take all graphic elements into account. The option is to choose how much the graphics will be simplified. In the given program, we chose the cover, which achieves maximum computing power. For graphically complex objects, it is not possible to consider a computational model and a real one. The model will be simplified for the calculation, the real model will be plotted and still get sufficiently accurate results.

10. Conclusions

We managed to design a graphical model of the labyrinth, the mouse, and we created all the necessary algorithms so that the mouse could move around the labyrinth according to the instructions. In the program we have implemented a function for identifying collision states in 3D space, which we described in more detail in the article. In the article, we also showed the visual processing of the labyrinth we created. It has 3 partitions that represent obstacles in space. Exterior walls are also obstacles. The mouse is imported as a stl file with 100% visualization. The model we have chosen has been simplified to the cover collision calculation, which is 4 areas around the micromouse.

The proposed principles can be extended and fine-tuned for use in other areas, tracer robots [17] and industrial robots [18].

Acknowledgements

This paper was developed within the project KEGA 006STU-4/2021: "Progressive form of interdisciplinary education and support for the development of vocational subjects in the university environment".

References

[1]. Arthaya, B., Sadiyoko, A., & Hadiwidjaja, A. (2006). The design of a maze solving system for a micromouse by using a potential value algorithm. *World Transactions on Engineering and Technology Education*, *5*(3), 509.

[2]. Auyeung, T. (2005). Micromouse handbook. *Pode consultar-se em*, 24-26.

[3]. Daneshjo, N., Kralík, M., Petrovčiková, K., Dudáš Pajerská, E., & Paštéka, M. (2019). Avoiding the Obstacles in the Robot Working Zone by using the Lee Algorithm. *Advances in Science and Technology. Research Journal*, *13*(2).

[4]. Božek, P., Bezák, P., Nikitin, Y., Fedorko, G., & Fabian, M. (2015). Increasing the production system productivity using inertial navigation. *Manuf. Technol*, *15*, 274-278.

[5]. Grabowska, S. (2020). Smart factories in the age of Industry 4.0. *Management systems in production engineering*, *28*(2), 90-96.

[6]. Vacheva, G., Hinov, N., & Gilev, B. (2020). Innovative Methods for Control of Electric Vehicles. *TEM Journal*, *9*(1), 14-18.

[7]. Su, Y. X., Zheng, C. H., Müller, P. C., & Duan, B. Y. (2006). A simple improved velocity estimation for low-speed regions based on position measurements only. *IEEE Transactions on Control Systems Technology*, *14*(5), 937-942.

[8]. Wang, D., Yu, X., Wan, W., & Xu, H. (2008, July). A new method of infrared sensor measurement for micromouse control. In *2008 International Conference on Audio, Language and Image Processing* (pp. 784-787). IEEE.

[9]. Wu, J., Cai, J., Fan, L., Huo, M., & Zheng, Z. (2013). Design and Algorithm Research of High Speed Two-wheeled Mini Robots. *Sensors & Transducers*, *25*, 189.

[10]. Lorenz, R. D. (2001). Robotics and automation applications of drives and converters. *Proceedings of the IEEE*, *89*(6), 951-962.

[11]. Kibler, S. G., Hauer, A. E., Giessel, D. S., Malveaux, C. S., & Raskovic, D. (2011, April). IEEE Micromouse for mechatronics research and education. In *2011 IEEE International Conference on Mechatronics* (pp. 887-892). IEEE.

[12]. Chen, N., Chung, H., & Kwon, Y. K. (1995). Integration of micromouse project with undergraduate curriculum: a large-scale student participation approach. *IEEE transactions on education*, *38*(2), 136-144.

[13]. Su, J. H., Huang, H. H., & Lee, C. S. (2013, December). Behaveior model simulations of micromouse and its application in intelligent mobile robot education. In *2013 CACS International Automatic Control Conference (CACS)* (pp. 511-515). IEEE.

[14]. Su, J. H., Cai, X. H., Lee, C. S., & Chen, C. W. (2016, August). The development of a half-size micromouse and its application in mobile robot education. In *2016 International Conference on Advanced Robotics and Intelligent Systems (ARIS)* (pp. 1-6). IEEE.

[15]. New Technology Foundation. (2011). Tsukuba, Japan, The 32nd All Japan Micromouse Contest. Retrieved from: https://www.ntf.or.jp/mouse/micromouse2011/index_EN.html [accessed: 10 September 2021].

[16]. Lozhkin, A., Maiorov, K., & Bozek, P. (2021). Convolutional neural networks training for autonomous robotics. *Management Systems in Production Engineering*, *29*(1), 75-79.

[17]. Pivarciova, E. L. E. N. A., & Csongrady, T. I. B. O. R. (2016). Tracer robot with a proportional control. *MM Sci. J*, 1277-1286.

[18]. Božek, P., Ivandić, Ž., Lozhkin, A., Lyalin, V., & Tarasov, V. (2016). Solutions to the characteristic equation for industrial robot's elliptic trajectories. *Teh Vjesn*, *23*(4), 1017-1023.