3D Simulator with Hardware-in-the-Loop capability for the Micromouse Competition

Luis Piardi*, Lucas Eckert†, José Lima*¶, Paulo Costa‡¶, Antonio Valente §¶, Alberto Nakano†
* Research Center in Digitalization and Intelligent Robotics (CeDRI),
Instituto Politécnico de Bragança, Bragança, Portugal
Email: {jllima, piardi}@ipb.pt
† Federal University of Technology - Paraná, Toledo, Brazil
Email: lucas.eck95@gmail.com, nakano@utfpr.edu.br
‡ Faculty of Engineering of University of Porto (FEUP), Porto, Portugal
Email: paco@fe.up.pt
§Universidade de Trás-os-Montes e Alto Douro (UTAD), Vila Real, Portugal
Email: avalente@utad.pt
¶ Centre for Robotics in Industry and Intelligent Systems - INESCTEC, Porto, Portugal

***Abstract*—Robotics competitions are a way to challenge researchers, roboticists and enthusiastic to address robot applications. One of the well-known international competition is the Micromouse where the fastest mobile robot to solve a maze is the winner. There are several topics addressed in this competition such as robot prototyping, control, electronics, path planning, optimization, among others while keeping the size of the robot as small as possible. A simulation can be used to speed-up the development and testing algorithms but faces the gap between a simulation and reality, specially in the dynamics behaviour. There are some simulation environments that allow to simulate the Micromouse competition, but in this paper, an Hardware-in-the-loop simulator tool is presented where the simulated robot is controlled by the same microcontroller used by the robot. By this way, the developed algorithms are tested and validated with the limitations and constraints presented in the real hardware, such as memory and processing capabilities. The robot dynamics, the slippage of the wheels, the friction and the 3D visualization are present in the simulator. The presented results show that the same code and hardware controlling the simulated and the real robot identically.**

***Index Terms*—Micromouse, Simulator, Robotics Competitions, Hardware in the Loop.**

I. Introduction

The technological advances that are taking place in this new century accelerate the use of robots and embedded systems in everyday life and even more widely in industry. However, before these complex systems are definitively propagate in society, the development team needs to perform various tests to validate the proposed system. Knowing that at this stage of development many test runs may fail and may even be destructive, the use of a simulation at this stage of the process is important and will speed up the process. In this context, the use of hardware in the loop (HIL) becomes a powerful tool to be used before the real tests and after the tests in an isolated simulation environment, replacing the electro-mechanical systems with a dynamic 3D simulator. Consequently, earlier in the process, software and hardware errors can be found.

978-1-7281-3558-8/19/$31.00 ©2019 IEEE

According to [1] HIL is a real-time simulation for embedded control systems, using real hardware and other control systems with a dynamic simulator that can replace the real system. To facilitate the understanding of HIL, and to verify where it lies between the real world and the simulation, see Figure 1. According to this Figure, when advancing in the axis of realization of tests in real conditions, the price, time and the risk increases. For example, for the real test of an airplane could result in an accident, being completely destructive, expensive and dangerous. On the other hand, using simulation alone will reduce costs, risks and time considerably, but the accuracy of the system also decreases. Therefore, HIL is at an intermediate stage, balancing costs with accuracy.

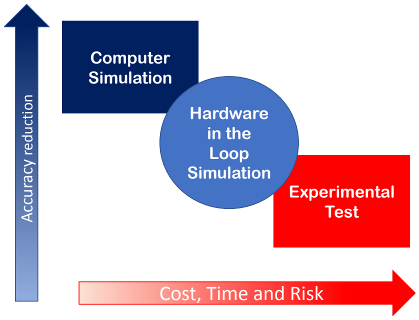


Fig. 1. Correlation of price, time, risk and precision between simulation and real systems. (adapted from [2]).

The Micromouse competition consists of a small autonomous mobile robot that, when placed in an unknown labyrinth, is able to map it, look for the best possible route between the starting point and the goal and travel it in the shortest possible time. To accomplish all these tasks the robot must be able to self-locate, map the labyrinth as it traverses and plan paths based on the map obtained. The method of self-localization is based on the odometry and distance sensors present in the robot, and on the previous knowledge of the start point and the general configuration of the environment. Several navigation methodologies, in order to cover the maximum possible terrain of the labyrinth while mapping, and route planning are analyzed for the best possible combination. The robot that locates the center of the labyrinth in a shorter time wins the competition.

The main contribution of this work is to present the development of a HIL for the Micromouse competition with a 3D simulator, in order to increase the accessibility of the competition and decrease the development time. This tool is available as open source, allowing each teams to make the adaptations they deem necessary in the scenario, robot and control algorithms.

An outline of this paper is as follows. In Section II a brief state of the art about HIL and Micromouse is presented. In Section III a description of the real and simulated robot is presented. Section IV presents the architecture of the HIL tool. Experimental results are presented in Section V and Section VI concludes the paper.

II. State of Art

In this chapter a brief state of the art reporting the HIL application in the robotics area will be presented. The benefits, developments and challenges in Micromouse competition will also be addressed.

A. HIL in Robotics Area

HIL systems have been widely applied because of their advantages during the process of development already mentioned in Section I. The main areas that have been developing and adopting HIL are aeronautical industries [3], [4], automotive industry [5]–[7] and power systems [8], [9].

Another field in evidence is robotics. HIL can be used in mobile robots and manipulators, which are constantly optimizing their performance. The work developed in [10], uses the HIL to test and validate the dynamic and cinematic control of a non holonomic wheeled mobile manipulators. In [11], it was proposed the development of HIL to simulate robotic manipulators with n degrees of freedom, considering the dynamism that occurs when a load is inserted in the system. In [12], it was used a mobile robot developed in simulation environment and controlled externally by an Arduino, to assist in the learning of laboratory classes of automatic control and robotics. The above examples show the usefulness of HIL in robotic systems. In order to facilitate the development of the prototypes and the validation of algorithms for mapping and path planning, is proposed in this work the elaboration of an HIL system directed to the Micromouse competition. All the source code and tools for this HIL simulation can be downloaded from [13].

B. Micromouse Competition

The Micromouse competition began in the late 1970s, being the first competition promoted by the IEEE. In this competition, 8 × 8 cells mazes were used and the robots achieve speed-run times around 30 seconds. Since then, Micromouse competitions have spread all over the world [14]. Nowadays, 16 x 16 cell mazes are used and the robots reach speed-run times of less than 7 seconds.

Although it started more than 35 years ago, the importance of the Micromouse problem in the field of robotics remains unparalleled, since it requires thorough analysis and proper planning to be solved [15]. In addition to the acquisition of technical skills, group competitions, such as the Micromouse, develop teamwork, time management and communication skills.

In the Micromouse competition, each robot begins with some basic knowledge of the maze, obtained from the rules [16]. The labyrinth has a square shape, consisting of 16 x 16 square cells of 18 cm x 18 cm, and has walls around the outer perimeter. These walls are 5 cm high, 1.2 cm wide and are painted white to reflect infrared light. The labyrinth floor is painted black to not reflect the infrared light. The initial position is one of the four corners of the maze and the goal is to reach the center. Other than that, it is necessary to explore and map the environment to perform the other tasks. In the presented paper, the focus is to present the developed HIL and simulation tool and not to address the optimized solution for the competition.

III. Robot Architecture and Driving Algorithm

In this section, the real robot developed for the competition Micromouse and which inspired the development of the simulated robot used in this work will be presented. The chapter concludes with an explanation of the standard motion wall follower algorithm.

A. Real Robot

The mobile robot shown in the Figure 2 was designed and developed for the purpose of completing a Micromouse challenge. Its structure is designed with dimensions that meet the rules of such competition [16], and with components that allow the robot to locate, move and identify the environment around it, necessary conditions to map the maze, plan the best way between the starting point and the goal and to go through this route.

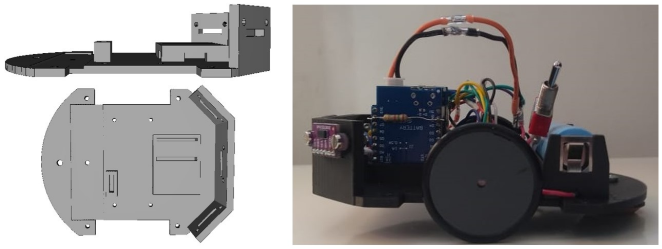


Fig. 2. Assembled Micromouse plate and robot.

The robot was steered using differential geometry. Therefore, the robot has three wheels, two of them connected to DC motors, while the other is a caster wheel that has the support function. This geometry was chosen due to the several advantages presented in comparison with the other architectures, highlighting: the ability to rotate without changing the position of its central axis, a requirement to maneuver in the narrow maze environments, mechanical and control simplicity, low maintenance rates and high odometric accuracy.

The Micromouse is designed to traverse a narrow maze at high speed. In order to achieve such performance, two factors have high importance: the robot weight must not be too high and its center of mass must lie within the triangle formed by its three wheels. To obtain such characteristics, a base plate has been specially developed, as shown in Figure 2.

The Micromouse robot has an elaborate electronic structure to move through the maze independently. To this end, the Micromouse incorporated in its structure some electronic components as batteries, a Wemos D1 mini (ESP8266 based microcontroller), Wemos shields, encoders, two DC motors and laser distance sensors (VL53L0XV2), positioned at the bumper with angles of 45, 90 and 135 degrees. Information about the robot's dimensions are presented in the Table I.

TABLE I
DIMENSIONS OF THE MICROMOUSE ROBOT

Robot Description	Dimension	Unit
Width	0.096	m
Length	0.120	m
Wheel diameter	0.032	m
Wheel thickness	0.008	m
Robot mass	1.25	kg

B. Simulated Robot

To obtain the results of the implementation, the SimTwo simulator was used. SimTwo is an open source realistic simulation software suitable for the design and development of solutions for several types of robots [17]. The dynamics realism is obtained by simulating each body and electric motor numerically using its physical characteristics: shape, mass and moments of inertia, surface friction and elasticity for the physical bodies and resistance, inductance, motor constant, rotor moment of inertia and several friction parameters for the DC motor [18].

The simulated Micromouse robot was assembled with the same dimensions as of the real one, presented at the Table I. The model is a combination of solids (cuboids and cylinders) and shells, elements without mass that do not modified the robot physical properties but are an essential part of the collisions simulation, connected by hinge joints (joints allows two objects to move between them through a single axis). Three distance sensors were installed in the same positions and angles as those present in the real robot. Connected to the side wheels are two DC motors whose simulation model can be checked in [18]. The resulting robot is presented in Figure 3.

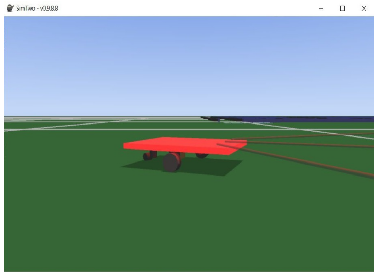


Fig. 3. Simulated Micromouse robot.

C. Driving Algorithm

The robot's control is obtained by the configuration of the speed to be applied in each motor. The set of such velocities defines the robot's final speed and the direction in which it moves. Two variables were used to control these characteristics. Such variables configure the angular velocity ω in $[rad/s]$ and linear velocity V $[m/s]$ to be performed by the robot. The constant b represents the distance between the traction wheels of the robot at its point of contact with the ground. The speed applied to each motor is defined according to the Equations (1) and (2).

$$V_{RightWheel} = V - \left(\omega \cdot \frac{b}{2}\right) \qquad [m/s] \tag{1}$$

$$V_{LeftWheel} = V + \left(\omega \cdot \frac{b}{2}\right) \qquad [m/s] \tag{2}$$

To ease the robot's control and avoid collisions during its movements, a centralization algorithm has been implemented. This algorithm uses the distance of the walls obtained by the laser distance sensors and the previous knowledge of the size of the cells to keep the robot centered around the currently unused axis. This centralization is realized when there is a wall near to the robot, as can be seen in Figure 4, and the correction is performed by setting the angular speed of the motors.

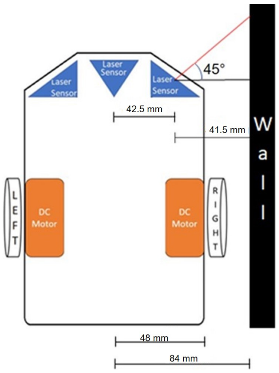


Fig. 4. Centralized robot measures.

To perform this correction a triangulation is applied. Using the known angle of the sensor (45 degrees), the size of the cell (16.8 × 16.8 cm) and the distance measured by the sensor, the current distance between the wall and the robot is calculated. This distance is then compared to the distance required for the robot to be in the center of the cell and the angular velocity is adjusted. The complete calculation of this variance is presented in the Equation (3), where x is the distance between the robot's sensor and the wall (41.5 mm). K is a constant that represents the proportional gain to cancel out the error of centralization of the robot in relation to the walls. The angular velocity (ω) is set according to the distance measured by the most significant sensor, i.e., if the sensor on the left side sensor has the largest distance component, the robot tends to turn left, else it turns right.

$$\omega = K \cdot \left(SensorDist - \left(\frac{x}{cos(45)} \right) \right) \qquad [rad/s] \qquad (3)$$

D. Wall Follower Algorithm

The most common algorithm for a maze solver robot is the wall follower algorithm, also called left hand/right hand rule. In this method, the robot will decide its direction by following the left or right wall. Whenever the robot arrives at a junction, it will detect the opening walls and select its direction, giving priority to the selected wall, in our case, the right wall. This selection occurs according to the following steps:

1) Sense the right wall.
2) If the right wall is not present, turn 90 degrees right and return to step 1.
3) Sense the front wall.
4) If the front wall is not present, move straight and return to step 1.
5) Sense the left wall.
6) If the left wall is not present, turn 90 degrees left, else rotate 180 degrees.
7) Return to step 1.

Taking the walls as a guide, this strategy is able to make the robot reach the goal of the maze without actually solving it. However, this algorithm is not an efficient method to solve a maze since the wall follower algorithm is not capable to solve mazes with a closed loop region [19]. Also remind that the main focus of this paper is to propose a HIL simulation tool,as described in the next section.

IV. Hardware in the Loop Tool

This chapter will present the architecture of hardware in the loop for the Micromouse competition. This tool provides a feature to test the hardware responsible for controlling all actions of the real robot, but controlling the virtual robot in SimTwo simulation environment.

A. System Architecture HIL

Figure 5 shows the implemented architecture system. Communication between the controller (Wemos) and the SimTwo runs in real time through USB port (Serial Communication) between devices. In other words, the simulator will provide the sensor data (right, left and front distance sensor and motors encoders) to the embedded controller in the hardware, which will process the data and control the actions of the virtual robot (right and left motors). The use of a real microcontroller stresses the typical hardware limitations such as memory and processing time.

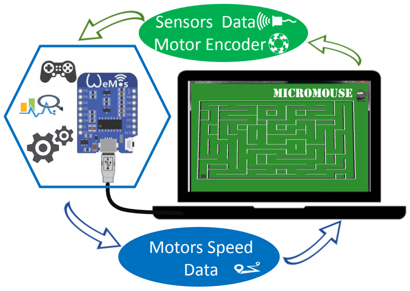


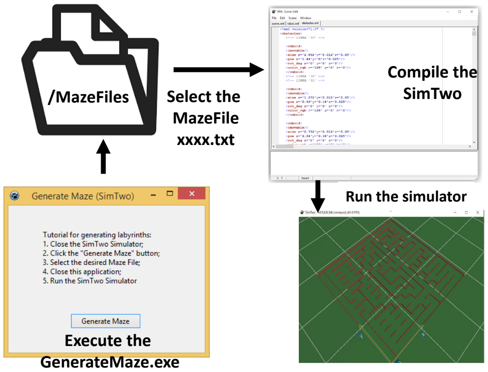
Fig. 5. Architecture HIL for Micromouse SimTwo simulator.

Once the serial port is used both for programming (Arduino IDE used for Wemos platform compilation) and for communicate with SimTwo, these steps below must be followed to execute the HIL:

1) Compile the code in the controller (Arduino IDE) and then close all dependencies of the Serial port (Monitor Serial);
2) On the SimTwo, in the $Config->I/O$ tab, configure the Serial port with a Baud Rate of 115200 [bps] and select the COM port corresponding to the Arduino IDE, and then open the communication. To finish this step, open the communication by selecting the "open" checkbox;
3) On the $Editor$ tab, compile the code ($Ctrl+F9$) and then execute pressing ($F9$). After this step the HIL tool will be in operation.

B. Maze Generator

Based on the necessity of validate and test the developed Micromouse algorithms in the most varied scenarios, a maze generator was implemented. It is an existing executable (MazeGenerator.exe) inside the SimTwo environment folder, which converts more than 450 different classic labyrinths encoded in a txt file (credits to [20]) to XML file "obstacle.xml" that will represent the maze in SimTwo simulator. Those labyrinths reproduce the competition environment with greater fidelity, having the same characteristics of a real competitive maze. The sequence to execute the MazeGenerator.exe application is described in Figure 6. All the developed tools for the Micromouse HIL can be downloaded from [13].


/MazeFiles
Select the MazeFile xxxx.txt
Compile the SimTwo
Run the simulator
Execute the GenerateMaze.exe
XML Scene Edit
File Edit Scene Window
scene.xml robot.xml obstacles.xml
Insert
SimTwo - v0.9.8.8 [Micromouse] (61.0 FPS)
Generate Maze (SimTwo)
Generate Maze

```
<?xml version="1.0" ?>
<obstacles>
  <!-- LINHA '33' -->

  <cuboid>
  <imovable/>
  <size x='2.892'y='0.012'z='0.05'/>
  <pos x='1.44'y='0'z='0.025'/>
  <rot_deg x='0' y='0' z='0'/>
  <color_rgb r='128' g='0' b='0'/>
  </cuboid>
  <!-- LINHA '32' -->
  <!-- LINHA '31' -->

  <cuboid>
  <imovable/>
  <size x='1.272'y='0.012'z='0.05'/>
  <pos x='0.99'y='0.18'z='0.025'/>
  <rot_deg x='0' y='0' z='0'/>
  <color_rgb r='128' g='0' b='0'/>
  </cuboid>

  <cuboid>
  <imovable/>
  <size x='0.732'y='0.012'z='0.05'/>
  <pos x='2.34'y='0.18'z='0.025'/>
  <rot_deg x='0' y='0' z='0'/>
```

Tutorial for generating labyrinths:
1. Close the SimTwo Simulator;
2. Click the "Generate Maze" button;
3. Select the desired Maze File;
4. Close this application;
5. Run the SimTwo Simulator

Fig. 6. Maze conversion process.

C. Methodology to Validate the Real Control Script

Before migrating the code to the robot and testing it in real environment, a sequence of three procedures are suggested to validate the control script. This proposal consists in performing tests on several mazes available in the simulator, subjecting the developed control code by the team to different conditions that can occur in a competition. This procedures is described in Figure 7. It consists in initially developing the code to the robot to perform exploration of the maze to find the best path between the initial position and the target position (procedure 1). The next step is to test the script in the HIL tool (procedure 2). If the test succeeds, replace the current maze (procedure 3) with a new maze and re-test your script. When encountering errors or new proposals to optimize the code, return to procedure 1. After a considerable number of tests occur between procedures 2 and 3, the script will be validated for testing in real environments within the real robot.

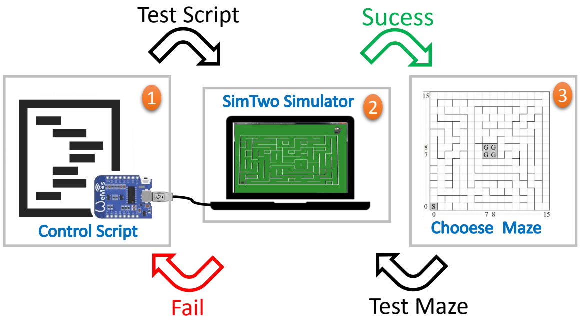


Fig. 7. Procedure to validate the control script.

V. Results

For the validation of HIL developed in this paper, this section will be separated into two steps. The first step will discuss the trajectory to explore the maze using the wall follower algorithm. The second step will present information obtained from the sensors provided by the simulator and speed information to be applied in the simulated robot provided by the real hardware control.

A. Map Exploration with Wall Follower Algorithm

The wall follower algorithm provided in this paper comes as standard to be compiled on the actual controller (e.g. Wemos D1 mini). The choice of this simple algorithm has the objective to foment the interest of new teams in the competition, being able to be a starting point for the development of more complex exploration algorithms particular of each team. Figure 8 presents the simulation environment with the maze hokurico-030-2012 (available in this tool), explored by the above mentioned algorithm. As predicted, the algorithm does not perform the complete exploration of the maze, initiating a cycle of movements every 140 seconds.

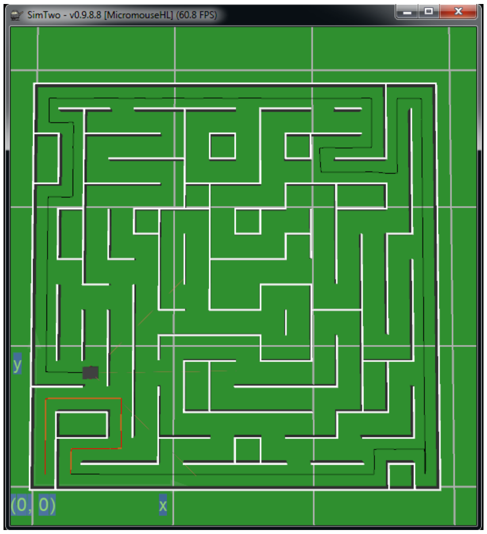


Fig. 8. Maze explored with wall follower algorithm.

B. Sensors and Speeds Motors Data

In the HIL tool the simulator sends the data acquired from the linked distance sensors in the simulated robot to the real controller (Wemos D1 mini). These sensors provide the distance in meters from the right, left and front walls. Figure 9 presents these data during the first 20 seconds of the scanning trajectory (red trail) presented in the Figure 8. These data are processed by the real controller and have the same magnitude of the sensors assembled in the real robot, due to the accurate model.

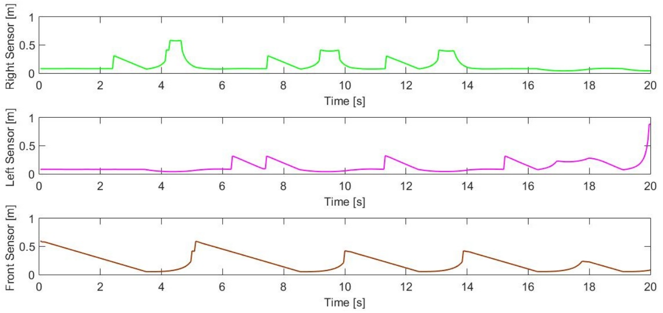


Fig. 9. Sensor distance data from SimTwo to Wemos platform.

The controller will process the simulator data in real time and returns the speeds of left and right wells to simulated robot. Figure 10 shows the speed values of the wheels during the first 20 seconds of the scanning trajectory presented in the Figure 8. The data of Figures 9 and 10 correspond to the same path and were acquired at the same time. Taking into consideration the high torque of the DC motor and the relatively small mass of the robot, the reference speed of the controller and the speed of the simulated robot are practically the same.

By correlating the data of the graphs, the following events can be observed:

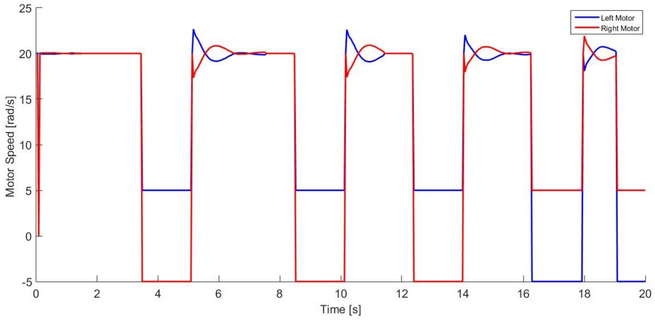


Fig. 10. Motor speed data from Wemos platform to SimTwo.

- Between the interval from 0 to 3.8 seconds, the right and left sensors are the same and constant values. The front sensor is decreasing linearly, i.e. the robot is centralized, with a constant speed of 20 $[rad/s]$ applied to both wheels. This will occur until the front sensor detects a measurement less than 0.1 $[m]$.
- At the moment 2.3 seconds a disturbance in the right sensor occurs due to the existence of a corner;
- During the interval from 3.8 to 5 seconds, the speed of the motors has the same intensity but in the opposite direction $\pm 5[rad/s]$, resulting in a rotation about its robot axis. When executing the rotation, the right sensor detects the halfway in the interval between 4 to 5 seconds.
- At intervals of 5 to 8 seconds a speed transition occurs, as the robot finishes the rotation around its axis and returns to straight way. Moments after the transition is observed a ripple at the speeds applied to the motor, because the robot is centralizing;
- The other events have behavior similar to those described above.

VI. Conclusion and Future Work

While the Micromouse competition started more than 35 years ago, it still is an important challenge to the researchers. It address robotics topics such as prototyping, control, localization, path planning, among the others. This paper described a Hardware-in-the-loop tool to this competition, where the microcontroller present in the robot can be used to control the simulation environment based on SimTwo. Using the same microcontroller and the same programming language, it is possible to test and fix the developed solutions faster and easier, even before the real robot is developed. The presented results showed the proposed tool (provided by the authors) can help the development of the Micromouse solution. As future work, the accuracy of the assumed models should be improved and other sensors can be modelled and included both in the Simulation and in the Hardware-in-the-loop communication protocol.

Acknowledgment

This work is financed by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

References

[1] P. Sarhadi and S. Yousefpour, "State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software," *International Journal of Dynamics and Control*, vol. 3, no. 4, pp. 470–479, 2015.

[2] J. L. Burbank, W. Kasch, and J. Ward, *An introduction to network modeling and simulation for the practicing engineer*, vol. 5. John Wiley & Sons, 2011.

[3] M. Montazeri-Gh and M. Nasiri, "Hardware-in-the-loop simulation for testing of electro-hydraulic fuel control unit in a jet engine application," *Simulation*, vol. 89, no. 2, pp. 225–233, 2013.

[4] M. Karpenko and N. Sepehri, "Hardware-in-the-loop simulator for research on fault tolerant control of electrohydraulic flight control systems," in *American Control Conference, 2006*, pp. 7–pp, IEEE, 2006.

[5] A. Palladino, G. Fiengo, and D. Lanzo, "A portable hardware-in-the-loop (hil) device for automotive diagnostic control systems," *ISA transactions*, vol. 51, no. 1, pp. 229–236, 2012.

[6] M. Lee, H. Lee, K. S. Lee, S. Ha, J. Bae, J. Park, H. Park, H. Choi, and H. Chun, "Development of a hardware in the loop simulation system for electric power steering in vehicles," *International journal of Automotive technology*, vol. 12, no. 5, p. 733, 2011.

[7] G. Naus, J. Ploeg, M. Van de Molengraft, W. Heemels, and M. Steinbuch, "Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach," *Control Engineering Practice*, vol. 18, no. 8, pp. 882–892, 2010.

[8] O. Crăciun, A. Florescu, I. Munteanu, A. I. Bratcu, S. Bacha, and D. Radu, "Hardware-in-the-loop simulation applied to protection devices testing," *International Journal of Electrical Power & Energy Systems*, vol. 54, pp. 55–64, 2014.

[9] M. O. Faruque and V. Dinavahi, "Hardware-in-the-loop simulation of power electronic systems using adaptive discretization," *IEEE transactions on industrial electronics*, vol. 57, no. 4, pp. 1146–1158, 2010.

[10] G. D. White, R. M. Bhatt, C. P. Tang, and V. N. Krovi, "Experimental evaluation of dynamic redundancy resolution in a nonholonomic wheeled mobile manipulator," *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 3, pp. 349–357, 2009.

[11] A. Martin and M. R. Emami, "Dynamic load emulation in hardware-in-the-loop simulation of robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 7, pp. 2980–2987, 2011.

[12] I. Tejado, J. Serrano, E. Pérez, D. Torres, and B. M. Vinagre, "Low-cost hardware-in-the-loop testbed of a mobile robot to support learning in automatic control and robotics," *IFAC-PapersOnLine*, vol. 49, no. 6, pp. 242–247, 2016.

[13] M. tools, "Github repository with developed tools for the micromouse competition." https://github.com/P33a/SimTwo/releases/tag/2019Jan, 2018.

[14] S. G. Kibler, A. E. Hauer, D. S. Giessel, C. S. Malveaux, and D. Raskovic, "Ieee micromouse for mechatronics research and education," in *Mechatronics (ICM), 2011 IEEE International Conference on*, pp. 887–892, IEEE, 2011.

[15] S. Mishra and P. Bande, "Maze solving algorithms for micro mouse," in *Signal Image Technology and Internet Based Systems, 2008. SITIS'08. IEEE International Conference on*, pp. 86–93, IEEE, 2008.

[16] M. P. Contest, "Rules of the micromouse portuguese competest."

[17] C. Paulo, G. José, L. José, and M. Paulo, "Simtwo realistic simulator: A tool for the development and validation of robot software," *Theory and Applications of Mathematics & Computer Science*, vol. 1, no. 1, pp. 17–33, 2011.

[18] J. Gonçalves, J. Lima, P. J. Costa, and A. P. Moreira, "Modeling and simulation of the emg30 geared motor with encoder resorting to simtwo: the official robot@ factory simulator," in *Advances in Sustainable and Competitive Manufacturing Systems*, pp. 307–314, Springer, 2013.

[19] M. R. Nepali, N. Yadav, D. A. H. Prasad, and S. Balasubramaniam, "A novel wall following algorithm for mobile robots," *International Journal of Robotics and Automation (IJRA)*, vol. 5, no. 2, p. 15, 2014.

[20] micromouseonline, "Github repository with maze files to micromouse." https://github.com/micromouseonline/mazefiles, 2018.

3D Simulator with Hardware-in-the-Loop capability for the Micromouse Competition

Por: Piardi, L (Piardi, Luis)[1]; Eckert, L (Eckert, Lucas)[2]; Lima, J (Lima, Jose)[1,5]; Costa, P (Costa, Paulo)[3,5]; Valente, A (Valente, Antonio)[4,5]; Nakano, A (Nakano, Alberto)[2]

Exibir ResearcherID e ORCID do Web of Science

2019 19TH IEEE INTERNATIONAL CONFERENCE ON AUTONOMOUS ROBOT SYSTEMS AND COMPETITIONS (ICARSC 2019)
Editado por: Almeida, L; Reis, LP; Moreira, AP
Série de livros: IEEE International Conference on Autonomous Robot Systems and Competitions ICARSC
Páginas: 110-115
Publicado: 2019
Tipo de documento: Proceedings Paper

Conferência

Conferência: 19th IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)

Local: Gondomar, PORTUGAL

Data: APR 24-26, 2019

Patrocinador(es): IEEE; Portuguese Soc Robot; IEEE Portugal Sect; EEE Robot & Automat Soc, Portuguese Chapter; Univ Porto, Fac Engn; Inst Syst & Comp Engn, Lab Assoc

Resumo

Robotics competitions are a way to challenge researchers, roboticists and enthusiastic to address robot applications. One of the well-known international competition is the Micromouse where the fastest mobile robot to solve a maze is the winner. There are several topics addressed in this competition such as robot prototyping, control, electronics, path planning, optimization, among others while keeping the size of the robot as small as possible. A simulation can be used to speed-up the development and testing algorithms but faces the gap between a simulation and reality, specially in the dynamics behaviour. There are some simulation environments that allow to simulate the Micromouse competition, but in this paper, an Hardware-in-the-loop simulator tool is presented where the simulated robot is controlled by the same microcontroller used by the robot. By this way, the developed algorithms are tested and validated with the limitations and constraints presented in the real hardware, such as memory and processing capabilities. The robot dynamics, the slippage of the wheels, the friction and the 3D visualization are present in the simulator. The presented results show that the same code and hardware controlling the simulated and the real robot identically.

Palavras-chave

Palavras-chave de autor: Micromouse; Simulator; Robotics Competitions; Hardware in the Loop

KeyWords Plus: ROBOT

Informaçoes sobre autores

Endereço de reprint: Lima, J (autor de reprint)

+ Inst Politecn Braganca, Res Ctr Digitalizat & Intelligent Robot CeDRI, Braganca, Portugal.

Endereço de reprint: Lima, J (autor de reprint)

Ctr Robot Ind & Intelligent Syst INESCTEC, Porto, Portugal.

Endereços:

+ [1] Inst Politecn Braganca, Res Ctr Digitalizat & Intelligent Robot CeDRI, Braganca, Portugal

+ [2] Univ Tecnol Fed Parana, Toledo, Brazil

+ [3] Univ Porto FEUP, Fac Engn, Porto, Portugal

+ [4] Univ Tras Os Montes & Alto Douro UTAD, Vila Real, Portugal

[5] Ctr Robot Ind & Intelligent Syst INESCTEC, Porto, Portugal

Endereços de e-mail: piardi@ipb.pt; lucas.eck95@gmail.com; jllima@ipb.pt; paco@fe.up.pt; avalente@utad.pt; nakano@utfpr.edu.br

Financiamento

Agência financiadora	Número do subsídio
ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Programme	POCI-01-0145-FEDER-006961
National Funds through the FCT Fundacao para a Ciencia e a Tecnologia (Portuguese Foundation for Science and Technology)	UID/EEA/50014/2013

Visualizar texto de financiamento

Editor

IEEE, 345 E 47TH ST, NEW YORK, NY 10017 USA

Categorias/Classificação

Áreas de pesquisa: Computer Science; Engineering; Robotics

Categorias do Web of Science: Computer Science, Artificial Intelligence; Engineering, Electrical & Electronic; Robotics

Ver mais campos de dados

Rede de citações

Em Web of Science Core Collection

0

Número de citações

Criar alerta de citação

18

Referências citadas

Visualizar Related Records

Usar no Web of Science

No Web of Science Contagem de Uso

1	1
Últimos 180 dias	Desde 2013

Saiba mais

Esse registro é de:
Web of Science Core Collection
- Conference Proceedings Citation Index - Science

Sugerir uma correção

Se quiser melhorar a qualidade dos dados deste registro, sugira uma correção.

Referências citadas: 18

Exibindo 18 de 18 **Visualizar todos na página Referências citadas** *(de Web of Science Core Collection)*

1. Title: [not available]
By: [Anonymous].
Rules of the micromouse portuguese competest
M. P. Contest
Times Cited: 1

2. Title: [not available]
By: Burbank, J. L.; Kasch, W.; Ward, J.
An introduction to network modeling and simulation for the practicing engineer Volume: 5 Published: 2011
Publisher: John Wiley & Sons
Times Cited: 3

3. **SimTwo Realistic Simulator: A Tool for the Development and Validation of Robot Software**
By: Costa, P.; Goncalves, J.; Lima, J.; et al.
Theory and Applications of Mathematics & Computer Science Volume: 1 Pages: 17-33 Published: 2011
[Show additional data]
Times Cited: 10

4. **Hardware-in-the-loop simulation applied to protection devices testing**
By: Craciun, Octavian; Florescu, Adrian; Munteanu, Iulian; et al.
INTERNATIONAL JOURNAL OF ELECTRICAL POWER & ENERGY SYSTEMS Volume: 54 Pages: 55-64 Published: JAN 2014
Times Cited: 16

5. **Hardware-in-the-Loop Simulation of Power Electronic Systems Using Adaptive Discretization**
By: Faruque, M. Omar; Dinavahi, Venkata
IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS Volume: 57 Issue: 4 Pages: 1146-1158 Published: APR 2010
Times Cited: 65

6. **Modeling and simulation of the emg30 geared motor with encoder resorting to simtwo: the official robot@ factory simulator**
By: Goncalves, J.; Lima, J.; Costa, P. J.; et al.
ADV SUSTAINABLE COMP Pages: 307-314 Published: 2013
Publisher: Springer
[Show additional data]
Times Cited: 5

7. **Hardware-in-the-loop simulator for research on fault tolerant control of electrohydraulic flight control systems**
By: Karpenko, M.; Sepehri, N.
2006 American Control Conference (IEEE Cat. No. 06CH37776C) Pages: 7 pp. Published: 2006
Times Cited: 1

8. **IEEE Micromouse for mechatronics research and education**
By: Kibler, S.G.; Hauer, A.E.; Giessel, D.S.; et al.
Proceedings of the 2011 IEEE International Conference on Mechatronics (ICM) Pages: 887-92 Published: 2011
Times Cited: 4

9. **Development of a hardware in the loop simulation system for electric power steering in vehicles**
By: Lee, M. H.; Lee, H. M.; Lee, K. S.; et al.
INTERNATIONAL JOURNAL OF AUTOMOTIVE TECHNOLOGY Volume: 12 Issue: 5 Pages: 733-744 Published: OCT 2011
Times Cited: 14

10. **Dynamic Load Emulation in Hardware-in-the-Loop Simulation of Robot Manipulators**
By: Martin, Adrian; Emami, M. Reza
IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS Volume: 58 Issue: 7 Pages: 2980-2987 Published: JUL 2011
Times Cited: 21

11. **Maze Solving Algorithms for Micro Mouse**
By: Mishra, Swati; Bande, Pankaj
SITIS 2008: 4TH INTERNATIONAL CONFERENCE ON SIGNAL IMAGE TECHNOLOGY AND INTERNET BASED SYSTEMS, PROCEEDINGS Pages: 86-+ Published: 2008
Times Cited: 16

12. **Hardware-in-the-loop simulation for testing of electro-hydraulic fuel control unit in a jet engine application**
By: Montazeri-Gh, M.; Nasiri, M.
SIMULATION-TRANSACTIONS OF THE SOCIETY FOR MODELING AND SIMULATION INTERNATIONAL Volume: 89 Issue: 2 Pages: 225-233 Published: FEB 2013
Times Cited: 8

13. **Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach**
By: Naus, G. J. L.; Ploeg, J.; Van de Molengraft, M. J. G.; et al.
CONTROL ENGINEERING PRACTICE Volume: 18 Issue: 8 Pages: 882-892 Published: AUG 2010
Times Cited: 41

14. **A novel wall following Algorithm for mobile robots**
By: Nepali, M. R.; Yadav, N.; Prasad, D. A. H.; et al.
International Journal of Robotics and Automation (IJRA) Volume: 5 Issue: 2 Pages: 15 Published: 2014
[Show additional data]
Times Cited: 2

15. **A portable hardware-in-the-loop (HIL) device for automotive diagnostic control systems**
By: Palladino, A.; Fiengo, G.; Lanzo, D.
ISA TRANSACTIONS Volume: 51 Issue: 1 Pages: 229-236 Published: JAN 2012
Times Cited: 17

16. **State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software**
By: Sarhadi, P.; Yousefpour, S.
Int. J. Dyn. Control Volume: 3 Pages: 470-479 Published: 2015
Times Cited: 16

17. **Low-cost Hardware-in-the-loop Testbed of a Mobile Robot to Support Learning in Automatic Control and Robotics**
Times Cited: 6

By: Tejado, Ines; Serrano, Juan; Perez, Enriliano; et al.
IFAC PAPERSONLINE Volume: 49 Issue: 6 Pages: 242-247 Published: 2016

18. Experimental Evaluation of Dynamic Redundancy Resolution in a Nonholonomic Wheeled Mobile Manipulator — Times Cited: 51
By: White, Glenn D.; Bhatt, Rajankumar M.; Tang, Chin Pei; et al.
IEEE-ASME TRANSACTIONS ON MECHATRONICS Volume: 14 Issue: 3 Pages: 349-357 Published: JUN 2009

Exibindo 18 de 18 Visualizar todos na página Referências citadas

Document details

1 of 1

Export Download Print E-mail Save to PDF Add to List More...

Serviços View at Publisher

19th IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2019
April 2019, Article number 8733640
19th IEEE International Conference on Autonomous Robot Systems and Competitions, ICARSC 2019; Porto; Portugal; 24 April 2019 through 26 April 2019; Category numberCFP1982X-ART; Code 148687

3D Simulator with Hardware-in-the-Loop capability for the Micromouse Competition (Conference Paper)

Piardi, L.[a], Eckert, L.[b], Lima, J.[a,e], Costat, P.[c,e], Valente, A.[d,e], Nakano, A.[b]

[a]Research Center in Digitalization and Intelligent Robotics (CeDRI), Instituto Politécnico de Bragança, Bragança, Portugal
[b]Federal University of Technology - Paraná, Toledo, Brazil
[c]Faculty of Engineering of University of Porto (FEUP), Porto, Portugal

View additional affiliations

Abstract

View references (20)

Robotics competitions are a way to challenge researchers, roboticists and enthusiastic to address robot applications. One of the well-known international competition is the Micromouse where the fastest mobile robot to solve a maze is the winner. There are several topics addressed in this competition such as robot prototyping, control, electronics, path planning, optimization, among others while keeping the size of the robot as small as possible. A simulation can be used to speed-up the development and testing algorithms but faces the gap between a simulation and reality, specially in the dynamics behaviour. There are some simulation environments that allow to simulate the Micromouse competition, but in this paper, an Hardware-in-the-loop simulator tool is presented where the simulated robot is controlled by the same microcontroller used by the robot. By this way, the developed algorithms are tested and validated with the limitations and constraints presented in the real hardware, such as memory and processing capabilities. The robot dynamics, the slippage of the wheels, the friction and the 3D visualization are present in the simulator. The presented results show that the same code and hardware controlling the simulated and the real robot identically. © 2019 IEEE.

SciVal Topic Prominence

Topic: Synthetic apertures | Power electronics | Power hardware-in-the-loop

Prominence percentile: 95.676

Author keywords

Hardware in the Loop | Micromouse | Robotics Competitions | Simulator

Indexed keywords

Engineering controlled terms:	Competition, Hardware-in-the-loop simulation, Motion planning, Robot applications, Robotics, Robots, Simulators, Synthetic apertures, Three dimensional computer graphics
Engineering uncontrolled terms	Development and testing, Hard-ware-in-the-loop, Hardware in the loop simulator, International competitions, Micromouse, Processing capability, Robotics competitions, Simulation environment
Engineering main heading:	Robot programming

Metrics View all metrics

PlumX Metrics

Usage, Captures, Mentions, Social Media and Citations beyond Scopus.

Cited by 0 documents

Inform me when this document is cited in Scopus:

Set citation alert

Set citation feed

Related documents

State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software
Sarhadi, P. , Yousefpour, S.
(2015) International Journal of Dynamics and Control

3D Simulator Based on SimTwo to Evaluate Algorithms in Micromouse Competition
Eckert, L. , Piardi, L. , Lima, J.
(2019) Advances in Intelligent Systems and Computing

Calibrate the turning of micromouse
Zhang, H.M. , Peh, L.S. , Wang, Y.H.
(2014) Advanced Materials Research

View all related documents based on references

Find more related documents in Scopus based on:

Authors Keywords

Funding details

Funding sponsor	Funding number	Acronym
China National Funds for Distinguished Young Scientists		
Fundação para a Ciência e a Tecnologia See opportunities by FCT↗		FCT
	UID/EEA/50014/2013	
	POCI-01-0145-FEDER-006961	
European Regional Development Fund		FEDER

Funding text

ACKNOWLEDGMENT This work is financed by the ERDF European Regional Development Fund through the Operational Programme for Competitiveness and Internationalisation - COMPETE 2020 Pro- gramme within project POCI-01-0145-FEDER-006961, and by National Funds through the FCT Fundac¸ão para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) as part of project UID/EEA/50014/2013.

ISBN: 978-172813558-8
Source Type: Conference Proceeding
Original language: English

DOI: 10.1109/ICARSC.2019.8733640
Document Type: Conference Paper
Volume Editors: Almeida L.,Reis L.P.,Moreira A.P.
Sponsors: IEEE,IEEE Portugal Section,IEEE Robotics and Automation Society (RAS),Portuguese Society for Robotics
Publisher: Institute of Electrical and Electronics Engineers Inc.

References (20)

View in search results format >

□ All Export Print E-mail Save to PDF Create bibliography

□ 1 Sarhadi, P., Yousefpour, S.

State of the art: hardware in the loop modeling and simulation with its applications in design, development and implementation of system and control software

(2015) *International Journal of Dynamics and Control*, 3 (4), pp. 470-479. Cited 24 times.
http://www.springer.com/engineering/mechanics/journal/40435/PS2
doi: 10.1007/s40435-014-0108-3

Serviços View at Publisher

□ 2 Burbank, J., Kasch, W., Ward, J.

An Introduction to Network Modeling and Simulation for the Practicing Engineer

(2011) *An Introduction to Network Modeling and Simulation for the Practicing Engineer*, pp. 1-203. Cited 14 times.
http://onlinelibrary.wiley.com/book/10.1002/9781118063651
ISBN: 978-111806365-1; 978-047046726-8
doi: 10.1002/9781118063651

Serviços View at Publisher

□ 3 Montazeri-Gh, M., Nasiri, M.

Hardware-in-the-loop simulation for testing of electro-hydraulic fuel control unit in a jet engine application

(2013) *Simulation*, 89 (2), pp. 225-233. Cited 12 times.
doi: 10.1177/0037549712466153

Serviços View at Publisher

□ 4 Karpenko, M., Sepehri, N.
Hardware-in-the-loop simulator for research on fault tolerant control of electrohydraulic flight control systems
(2006) *American Control Conference*, p. 7. Cited 2 times.
IEEE 2006

□ 5 Palladino, A., Fiengo, G., Lanzo, D.
A portable hardware-in-the-loop (HIL) device for automotive diagnostic control systems
(2012) *ISA Transactions*, 51 (1), pp. 229-236. Cited 24 times.
http://www.elsevier.com/wps/find/journaldescription.cws_home/524244/description#description
doi: 10.1016/j.isatra.2011.10.009
Serviços View at Publisher

□ 6 Lee, M.H., Lee, H.M., Lee, K.S., Ha, S.K., Bae, J.I., Park, J.H., Park, H.G., (...), Chun, H.H.
Development of a hardware in the loop simulation system for electric power steering in vehicles
(2011) *International Journal of Automotive Technology*, 12 (5), pp. 733-744. Cited 26 times.
doi: 10.1007/s12239-011-0085-x
Serviços View at Publisher

□ 7 Naus, G.J.L., Ploeg, J., Van de Molengraft, M.J.G., Heemels, W.P.M.H., Steinbuch, M.
Design and implementation of parameterized adaptive cruise control: An explicit model predictive control approach
(2010) *Control Engineering Practice*, 18 (8), pp. 882-892. Cited 58 times.
doi: 10.1016/j.conengprac.2010.03.012
Serviços View at Publisher

□ 8 Crăciun, O., Florescu, A., Munteanu, I., Bratcu, A.I., Bacha, S., Radu, D.
Hardware-in-the-loop simulation applied to protection devices testing
(2014) *International Journal of Electrical Power and Energy Systems*, 54, pp. 55-64. Cited 24 times.
doi: 10.1016/j.ijepes.2013.06.031
Serviços View at Publisher

□ 9 Omar Faruque, M.O., Dinavahi, V.
Hardware-in-the-loop simulation of power electronic systems using adaptive discretization
(2010) *IEEE Transactions on Industrial Electronics*, 57 (4), art. no. 5340670, pp. 1146-1158. Cited 90 times.
doi: 10.1109/TIE.2009.2036647
Serviços View at Publisher

□ 10 White, G.D., Bhatt, R.M., Tang, C.P., Krovi, V.N.
Experimental evaluation of dynamic redundancy resolution in a nonholonomic wheeled mobile manipulator
(2009) *IEEE/ASME Transactions on Mechatronics*, 14 (3), pp. 349-357. Cited 66 times.
doi: 10.1109/TMECH.2008.2008802
Serviços View at Publisher

□ 11 Martin, A., Emami, M.R.
Dynamic load emulation in hardware-in-the-loop simulation of robot manipulators
(2011) *IEEE Transactions on Industrial Electronics*, 58 (7), art. no. 5560807, pp. 2980-2987. Cited 28 times.
doi: 10.1109/TIE.2010.2072890
Serviços View at Publisher

☐ 12 Tejado, I., Serrano, J., Pérez, E., Torres, D., Vinagre, B.M.

Low-cost Hardware-in-the-loop Testbed of a Mobile Robot to Support Learning in Automatic Control and Robotics (Open Access)

(2016) *IFAC-PapersOnLine*, 49 (6), pp. 242-247. Cited 10 times.
http://www.journals.elsevier.com/ifac-papersonline/
doi: 10.1016/j.ifacol.2016.07.184

Serviços View at Publisher

☐ 13 Tools, M.
(2018) *Github Repository with Developed Tools for the Micromouse Competition*
2019Jan
https://github.com/P33a/SimTwo/releases/tag/

☐ 14 Kibler, S.G., Hauer, A.E., Giessel, D.S., Malveaux, C.S., Raskovic, D.

IEEE Micromouse for mechatronics research and education

(2011) *2011 IEEE International Conference on Mechatronics, ICM 2011 - Proceedings*, art. no. 5971240, pp. 887-892. Cited 23 times.
ISBN: 978-161284983-6
doi: 10.1109/ICMECH.2011.5971240

Serviços View at Publisher

☐ 15 Mishra, S., Bande, P.

Maze solving Algorithms for micro mouse

(2008) *SITIS 2008 - Proceedings of the 4th International Conference on Signal Image Technology and Internet Based Systems*, art. no. 4725791, pp. 86-93. Cited 52 times.
ISBN: 978-076953493-0
doi: 10.1109/SITIS.2008.104

Serviços View at Publisher

☐ 16 Contest, M.P.
Rules of the Micromouse Portuguese Competest

☐ 17 Paulo, C., José, G., José, L., Paulo, M.
Simtwo realistic simulator: A tool for the development and validation of robot software
(2011) *Theory and Applications of Mathematics & Computer Science*, 1 (1), pp. 17-33. Cited 23 times.

☐ 18 Gonçalves, J., Lima, J., Costa, P.J., Moreira, A.P.

Modeling and simulation of the EMG30 geared motor with encoder resorting to simtwo: The official robot@factory simulator

(2013) *Lecture Notes in Mechanical Engineering*, 7, pp. 307-314. Cited 9 times.
www.springer.com/series/11236
doi: 10.1007/978-3-319-00557-7_25

Serviços View at Publisher

☐ 19 Nepali, M.R., Yadav, N., Prasad, D.A.H., Balasubramaniam, S.
A novel wall following algorithm for mobile robots
(2014) *International Journal of Robotics and Automation (IJRA)*, 5 (2), p. 15. Cited 3 times.

☐ 20 (2018) *Github Repository with Maze Files to Micromouse*. Cited 2 times.
micromouseonline
https://github.com/micromouseonline/mazefiles

^ Top of page