Design and Implementation of a Real-Time Autonomous Navigation System Applied to Lego Robots

Thi Thoa Mac [*,**] **Cosmin Copot** [***] **Clara M. Ionescu** [*]

[*] *Dynamical Systems and Control Research Group, Department of Electrical Energy, Metals, Mechanical Constructions, and Systems, Ghent University, Ghent B-9052, Belgium (e-mail: {thoa.macthi, ClaraMihaela.Ionescu}@ugent.be).*
[**] *School of Mechanical Engineering, Hanoi University of Science and Technology, No.1 Dai Co Viet, Vietnam*
[***] *Department of Electromechanics, Op3Mech, University of Antwerp, Groenenborgerlaan 171, 2020 Antwerp, Belgium (email: Cosmin.Copot@uantwerpen.be)*

Abstract: Teaching theoretical concepts of a real-time autonomous robot system may be a challenging task without real hardware support. The paper discusses the application of the Lego Robot for teaching multi interdisciplinary subjects to Mechatronics students. A real-time mobile robot system with perception using sensors, path planning algorithm, PID controller is used as the case to demonstrate the teaching methodology. The novelties are introduced compared to classical robotic classes: (i) the adoption of a project-based learning approach as teaching methodology; (ii) an effective real-time autonomous navigation approach for the mobile robot. However, the extendibility and applicability of the presented approach are not limited to only the educational purpose.

Keywords: Education, LEGO MINDSTORMS, autonomous navigation, PID control, Embedded systems.

1. INTRODUCTION

The design of automatic machines and robots is required advanced knowledge from mechanics, automatic control and computer science. In high-level education it is therefore desirable to organize activities to let students acquire more powerful competencies for the integrated design of automatic machines and real-time, object-oriented control software. Recently, Lego Mindstorms, a commercial robot kit has been used for testing robotics applications (Rodriguez et al., (2016)) and more actively implemented as the teaching tool for mechatronics and control engineering (Kim et al., (2011)). It allows students to rapidly develop real-time robot applications. Lego Mindstorms have been already used within the framework of programming and control for robot competitions (Grandi et al., (2014)). The stable real-time control of a segway-like robot with a PID controller is used as case study to clarify the theoretical concepts. Zdesar et al., (2017) presents the teaching approach used at Autonomous Mobile Robots course that is about algorithms in autonomous navigation. In this area of technology, there is a gap between the researchers working on theoretical aspects and engineering working on problems with mechanical design, motor control, camera, sensors, computer, communication, etc. Therefore, project-based approach is a good option as it provides to students necessary skills for future engineering career.

In Chevalier et al., (2017)), a solution implemented at our lab to address the problem of practice for large groups of students is using remote lab applications. By this applications, the students are able to process the experiments from various locations. In Copot et al., (2017), we introduced the academic context, and projects with the course of ICT and Mechatronics (the full description of the course is available online (ICT and Mechatronics (2017)) at Ghent university such as (1) design of an Anti lock Braking System (ABS); (2) two-track-line follower system which is able to avoid obstacles present along the line; (3) sensor fusion on a two-track system; (4) design and implementing control strategy for a robot arm.

As autonomous navigation is one of the most important requirements of an intelligent vehicle, last year, we included this topic for the course of ICT and Mechatronics. The aim of this project is to design and implement a mobile robot that is able to safely reach the target while avoiding uncertain obstacles in unknown lab-scale environment. There are four required components of autonomous navigation: i) *perception*, the robot uses its sensors to extract meaningful information; ii) *localization*, the robot determines its location in the working space; iii) *cognition and path planning*, the robot decides how to steer to achieve its

[⋆] Sponsor and financial support acknowledgment goes here. Paper titles should be written in uppercase and lowercase letters, not all uppercase.

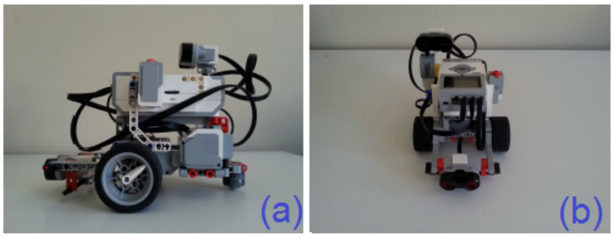

Fig. 1. The Lego robot design, (a) side view of the robot; (b) the front view of the robot.

goal; iv) *motion control*, the robot regulates its motion to accomplish the desired trajectory (Mac et al., (2016); Mac, (2016)). Therefore, in this project, students have to design mobile robot, choose appropriate sensor perception, develop suitable path planning and controller. The following innovative aspects are introduced with respect to classical robotic classes: (*i*) the use of Lego robots as teaching tools; (*ii*) the adoption of a project-based learning method as teaching methodology; (*iii*) providing an effective real-time autonomous navigation approach for mobile robot.

The remainder of this paper is structured as follows. In section 2, a brief introduction of mobile robot platform and obstacle detection method are presented. Section 3 describes the path planning algorithm while section 4 presents PID controller. The experiment is demonstrated in section 5, follow by section 6, where the main outcomes of this work are summarized and the future works are discussed.

2. THE MOBILE ROBOT PLATFORM AND OBSTACLE DETECTION METHOD

In this section, we introduce about robot design system and obstacles detection of a mobile platform.

2.1 Robot design

The robot design is shown in Fig 1 with two driven wheels. The third supporting wheel is a non-driven castor that follows the movement imposed by the two driven wheels. The wheel diameter is 5.5 cm. Both wheels are driven by EV3 Large Servo Motors with tacho feedback. In front of the robot, an ultrasonic sensor is mounted. The EV3 ultrasonic sensor can measure distances by emitting and detecting sound waves. It has a measuring range from 1 to 250 cm with an accuracy of ± 1 cm. On the left side of the brick, an EV3 Medium Servo Motor is mounted, with on top an EV3 Infrared Sensor. This composition makes it possible to rotate the infrared sensor 360^0. This sensor is able to detect an infrared signal emitted by a remote control infrared beacon. The EV3 intelligent brick is the heart and brain of the robot. It supports USB, Bluetooth and Wi-Fi communication with a computer.

2.2 Detecting obstacles using ultrasonic sensor

The robot has to plan a path to a destination point, so it needs to know where the obstacles are. For detecting the obstacles, an ultrasonic sensor is chosen because this sensor measures the distances accurately. The sensor is placed close to the ground, in this way even very low obstacles can be detected.

At the starting point the robot turns a certain angle and meanwhile measures the distances to the nearest object using the ultrasonic sensor. In this way the robot can detect all the obstacles around it. The data from the ultrasonic sensor and the angle at which this distance is measured can be transformed to $(x,\ y)$ coordinates. The algorithm

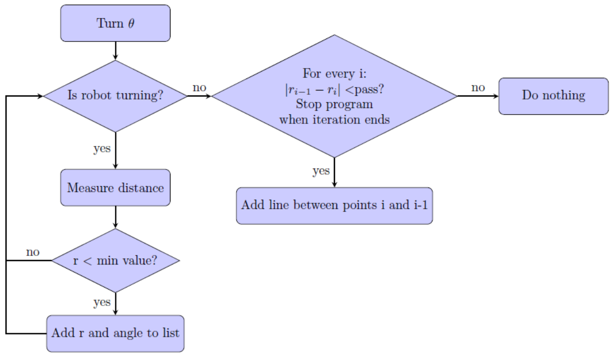

Fig. 2. Algorithm of detecting obstacles

for detecting obstacles is depicted in Fig 2, where θ is the total angle the robot turns while scanning and r is the distance measured by the sensor. The obstacle lines are then made into rectangles using the *LineToRectangle* class.

The way of detecting obstacles and making the obstacle lines is shown in Fig 3 when robot turns for θ degrees, while measuring every $\Delta\theta$. The ultrasonic sensor (red) in front of the robot performs measurements (blue dashed lines) and stores the calculated obstacle points (yellow dots) as a (x, y) coordinates. Obstacle points that are separated a distance Δr in radius smaller than a given value, are connected to obstacle lines (yellow lines connecting obstacle points). Those yellow lines will later be converted to rectangles as shown in Fig 4. At the left, the obstacles (green) with real dimensions are shown. Taking into account the robot dimension, the obstacles are extended as shown in the right.

3. PATH PLANNING ALGORITHM

For this project, the students are advised to choose one basic path planning method such as A*, D*, or shortest

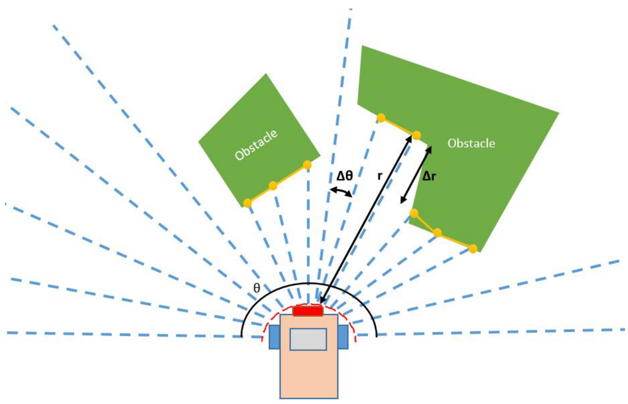

Fig. 3. Object detection and interpretation

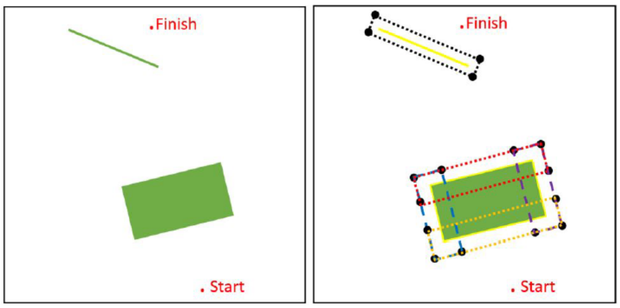

Fig. 4. Visualisation of LineMap formation

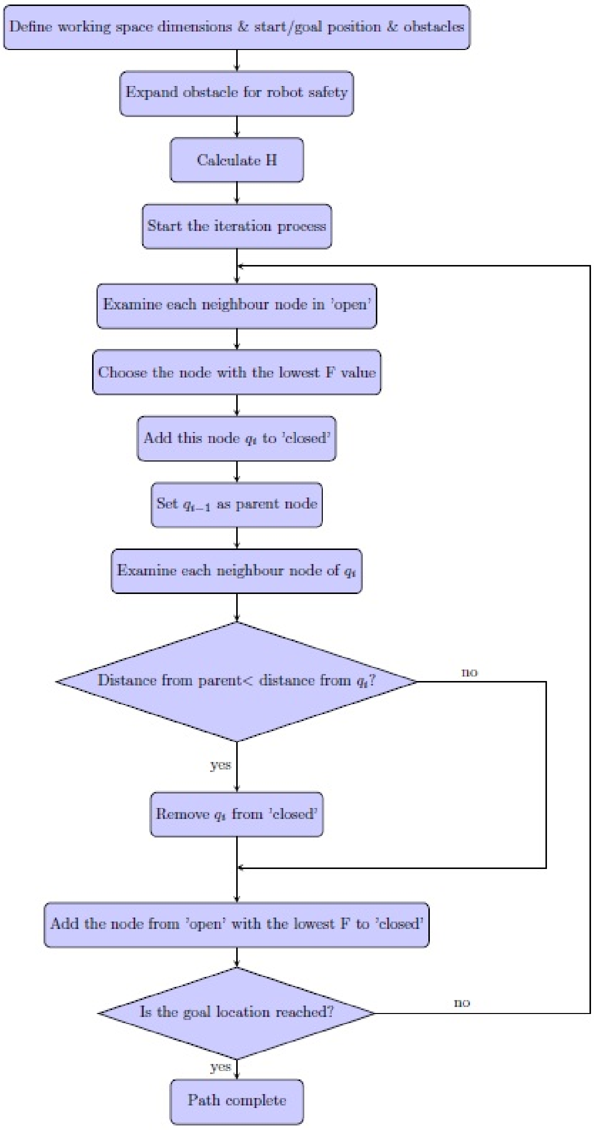

Fig. 5. Shortest path finder algorithm flowchart.

path finder method. In this section, we introduce about shortest path finder method which used in the leJOS programming language.

Similar to the A* algorithm, the path is found by searching among all possible paths to the target the one that has smallest cost. The cost function of the shortest finder method is given in (1)

$$F = G + H \tag{1}$$

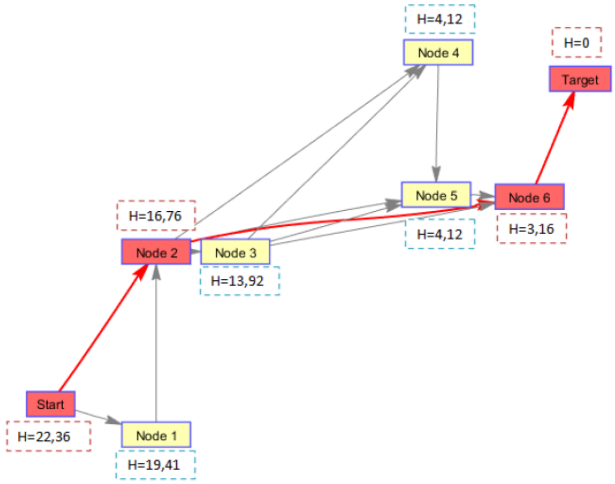

Fig. 6. The cost function graph of the shortest path finder method

where G is the cost from the start node the present node. H is the heuristic that estimate the cost of the cheapest path from that node to the goal. The main difference is that shortest path finder does not use a grid but uses its start and goal location, and the end points of all obstacles to construct a graph in the map. The algorithm is presented in Fig 5.

To show the concept, one example is depicted in Fig 6. From the start location, the possible paths are from *start* to node 1 or from *start* to node 2. The F-value from start to node 1 is calculated as the sum of its G-value (see Table 1), and the H-value of node 1 (as shown in Fig 6).

Table 1. G-values for all possible edges

Path	G - value
$start \to 1$	4,410
$start \to 2$	7,210
$1 \to 2$	7,000
$2 \to 3$	4,000
$2 \to 4$	13,42
$2 \to 5$	12,65
$2 \to 6$	13,60
$3 \to 4$	10,82
$3 \to 5$	9,850
$3 \to 6$	10,77
$4 \to 5$	1,000
$6 \to target$	3,160

Table 2. The F cost value

Path	F -value
$start \to 1$	23,82
$start \to 2$	23,97
$start \to 1 \to 2$	28,38
$start \to 2 \to 3$	25,13
$start \to 2 \to 4$	24,75
$start \to 2 \to 5$	23,98
$start \to 2 \to 5$	23,97
$start \to 2 \to 6 \to target$	23,97

As the F-value for the path between *start* and node 1, is lower than the one between *start* and node 2, node 1 is chosen for being added to the *closed* list. Next only one path is possible starting from node 1, namely the path to node 2. But similarly as explained for the A* algorithm,

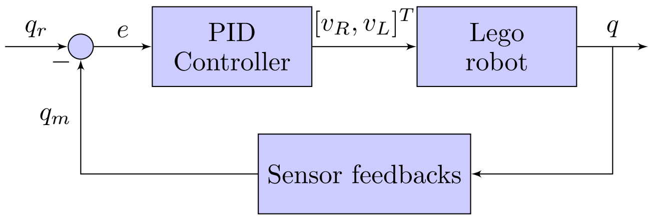

Fig. 7. Mobile robot control system.

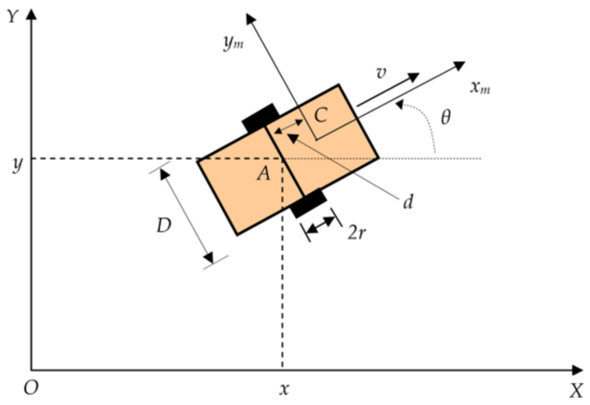

Fig. 8. The Kinematic model of the Lego robot.

the F-value when traveling directly from *start* to node 2 is lower than the one when traveling along the combined path *start*-node 1-node 2, as indicated in Table 2. So consequently node 1 is deleted of *closed* and is replaced by node 2.

Next, four paths are possible for the robot to travel with node 2 as a starting point namely from node 2 to node 3, node 4, node 5 and to node 6. The F-values for these paths are again given in Table 2. As can be seen in the table, the next node that will added to the *closed* list will be node 6, as the F-value from node 2 to node 6 is the lowest of the four paths starting from node 2. Finally the only path possible starting from node 6 is the path to the target location, and with that making an end to the shortest path search.

4. MOBILE ROBOT CONTROL SYSTEM

To this day, over 95 percentage of industrial applications are predominantly controlled by (PID) controllers due to is simplification and accuracy. A PID controller includes:

$$G_{PID}(s) = K_p + \frac{K_i}{s} + K_d s \tag{2}$$

where K_p, K_i, K_d denote the coefficients for the proportional, integral, and derivative terms. The proposed control system for the Lego robot in this study is shown in Fig 7. The control system consists of a PID controller, the kinematic model of mobile robot, a reference trajectory generator and two encoders which provides odometric information. In which v_L and v_R are the speed of the left and right wheel in cm/s respectively. Input, output, measured parameters are vectors: $q_r = [x_r, y_r, \theta_r]^T$, $q = [x, y, \theta]^T$, $q_m = [x_m, y_m, \theta_m]^T$. A schematic of the kinematic model is shown in Fig 8. The kinematic model is presented in equation (3). The parameters D and r are respectively the distance between two wheels and the wheel radius.

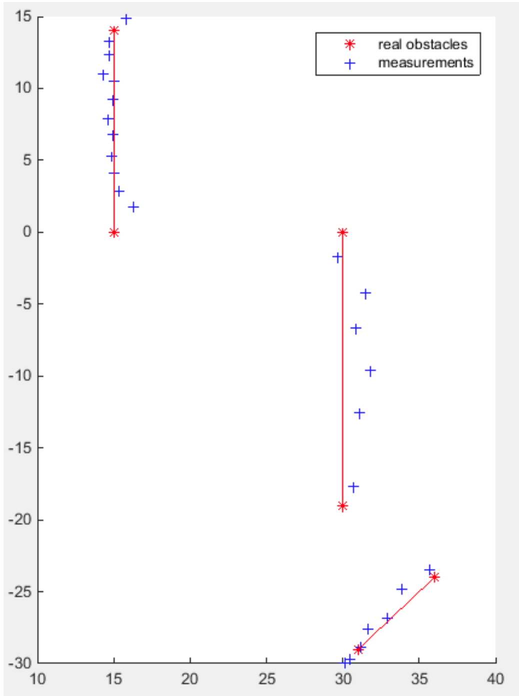

Fig. 9. Scanning unknown obstacles with ultrasonic sensor

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} r\cos\theta & r\cos\theta \\ r\sin\theta & r\sin\theta \\ \frac{r}{D} & -\frac{r}{D} \end{bmatrix} \begin{bmatrix} v_R \\ v_L \end{bmatrix} \tag{3}$$

5. EXPERIMENTAL RESULTS

5.1 Sensor accuracy

To decide which sensor to use to measure distances, an experiment has been performed. Both the ultrasonic sensor and the infrared sensor were tested. A comparison between real and measured values is shown in Table 3. It is clear that the ultrasonic sensor is the most accurate one: $\pm 1cm$. The measured distances of the infrared sensor seems to be different depending on the material of the object measured. Therefore, the ultrasonic sensor is used for obstacle detection in this project.

Table 3. The Distance Measurement

Real Distance	Infrared sensor black	Infrared sensor color	Ultrasonic Sensor
0	0	0	0
15	18	24	16.9
30	34	40	30.2
45	44	48	45.6
60	51	∞	60.2
75	∞	∞	75.1
90	∞	∞	91.1
105	∞	∞	105.0
120	∞	∞	120.1

The algorithm used to detect unknown obstacles, is described in Section 2. In the lab-scale environment, three known obstacles were placed around the robot and their actual positions were compared to the positions measured by the robot's sensor. Fig 10 shows the results of this test. These results are satisfactory, as the measured position is very close to the real obstacle.

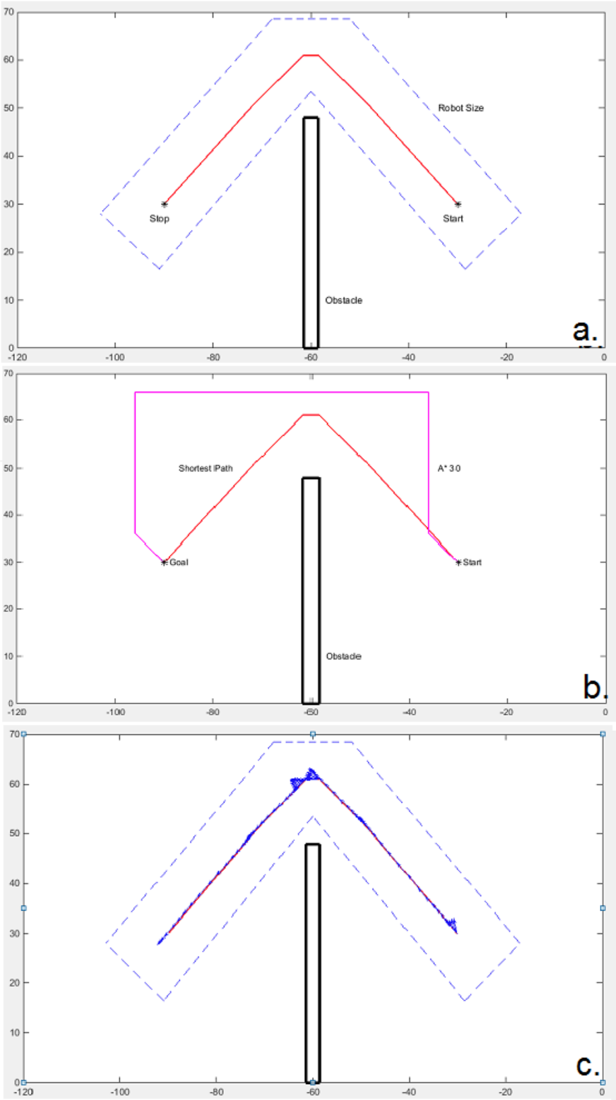

Fig. 10. (a.) Optimal path calculated by Shortest Path Finder method taking into account the robot size - red path; (b.) compare with the path obtained by A* algorithm - pink path; (c.) Real path experiment by Shortest Path Finder method.

5.2 *Shortest path finder experiment*

In this experiment, we evaluated the shortest path finder method by simulation and experiment. We also compared it with A* algorithm. Fig 10 presents the path obtained by shortest path finder algorithm - red path (a.); this path is compared to the one obtained by A* algorithm - pink path (b.) and the real path obtained by experiment - blue path (c.). The shortest path finder algorithm's path is shorter than A* algorithm's path. Moreover, the $ShortestPathFinder$ only uses the endpoints of obstacles as nodes, it has a much shorter calculation time, as the amount of endpoints of obstacles is usually much smaller than the amount of cells in a grid used in A* algorithm. It is also able to pass closer to obstacles than A*, as A* uses the middle of a grid cell as a way-point. This is a reason the $ShortestPathFinder$ algorithm is suggested in this project. Fig. 10 c shows that the robot is able to track the designed path accurately.

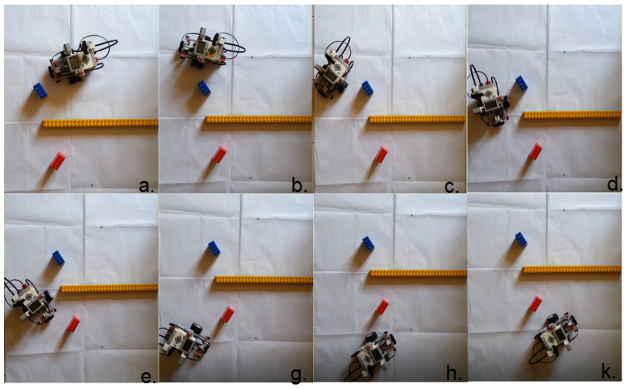

Fig. 11. Landing in presence of known/unknown obstacles results in the sequence of time

5.3 *Autonomous navigation experiment*

In this experiments, Lego Mindstorms EV3 was moving in presence of uncertain obstacles. The yellow long wall is a known obstacle, while the red and blue walls are unknown obstacles. The start point and target point are marked on the map. Experiments were carried out to test the ability of the robot performing autonomous task. As soon as the robot detected unknown obstacle by ultrasonic sensor, the position of that obstacle is provided. The results show that the Lego robot avoid all obstacles on its path and stop on the target position. The intended accuracy has been achieved using an equipped ultrasonic sensor.

6. CONCLUSION

The design of automatics machine and robot is a discipline that needs practical work at the learning process. To carry out it at the laboratory with the students, the LEGO system is proposed. This platform provides basically 2 advantages: its low price and good features. This study proposed an effective educational hands-on testbed so that students can learn a concept and design process for using sensor system, path planning, and control processing. This aim of the added project is making an autonomous robot that is able to drive to a goal while avoiding known and unknown obstacles.

In paper, we introduced $ShortestPathFinder$ algorithm, which is a modified version of the A* algorithm. This comparison results showed the $ShortestPathFinder$ algorithm to be the superior to the A* algorithm. The ultrasonic sensor was implemented to detect unknown obstacles, as it measures distances very accurately. Detecting unknown obstacles and determining their size was also done by using the ultrasonic sensor. The Lego robot was able to autonomously navigate in an unknown environment. The results of the experiments were satisfying.

The achievements of our teaching method compared to classical robotic classes are: (i) the applications of a project-based learning approach as teaching methodology; (ii) the development of an effective real-time autonomous navigation approach for mobile robot.

REFERENCES

Rodriguez, C., Guzman, J. L., Berenguel, M., Dormido, S. (2016), Teaching real-time programming using mobile robots, *IFAC-PapersOnLine vol. 49 (6), pp. 10-15.*

Kim, S., Oh, H., Kolaman, A., Tsourdos, A., White, B., Guterman, H. (2011), Educational hands-on test bed using Lego robot for learning guidance, navigation, and control, *IFAC Proceedings Volumes, vol. 44(1), pp. 5170-5175.*

Grandi, R., Falconi, R., Melchiorri, (2014), Robotic Competitions: Teaching Robotics and Real-Time Programming with LEGO Mindstorms, *IFAC Proceedings Volumes, vol. 47(3), pp. 10598-10603.*

Zdesar, A., Blazic, S., Klancar, G. (2017), Engineering Education in Wheeled Mobile Robotics, *IFAC Proceedings Volumes, vol. 50(1), pp. 12173-12178.*

Chevalier, A., Copot, C., Ionescu, C., DeKeyser, R. (2017), A three-year feedback study of a remote laboratory used in control engineering studies, *IEEE Transactions on Education, vol. 60(2), pp.127-133.*

Copot, C., Ionescu, C., DeKeyser, R. (2017), Interdisciplinary project-based learning at master level: control of robotic mechatronic *IFAC-PapersOnLine, vol. 49(6), pp. 314319.*

ICT and Mechatronics course content, Ghent University, *https://studiegids.ugent.be/2016/EN/studiefiches/E019330.pdf*

Mac, T. T., Copot, C., Tran, D .T., De Keyser, R. (2016), Heuristic approaches in robot path planning: a surveys, *Autonomous and Robotics System, vol. 86, pp. 13-28.*

Mac, T. T., Copot, C., Hernandez, A., De Keyser, R. (2016), Improved potential field method for unknown obstacle avoidance using UAV in indoor environment, *IEEE Intentional Symposium on Applied Machine Intelligence and Informatics, Herlany, Slovakia, pp. 345-350.*