

TERAKREDITASI RISTEKDIKTI No. 36b/E/KPT/2016

---

# Jurnal *Rekayasa Elektrika*

---

VOLUME 16 NOMOR 1

April 2020

---

**Autonomous Mobile Robot based on Behaviour-Based Robotic using V-REP Simulator–Pioneer P3-DX Robot** 15-22

*Esa Apriaskar, Fahmizal, Ika Cahyani, and Afrizal Mayub*

---

JRE	Vol. 16	No. 1	Hal 1–56	Banda Aceh, April 2020	ISSN. 1412-4785 e-ISSN. 2252-620X
-----	---------	-------	----------	---------------------------	--------------------------------------

# Autonomous Mobile Robot based on Behaviour-Based Robotic using V-REP Simulator–Pioneer P3-DX Robot

Esa Apriaskar<sup>1</sup>, Fahmizal<sup>2</sup>, Ika Cahyani<sup>2</sup>, and Afrizal Mayub<sup>3</sup>

<sup>1</sup>Department of Electrical Engineering, Universitas Negeri Semarang  
Gunungpati, Semarang 50229

<sup>2</sup>Department of Electrical Engineering and Informatics, Vocational College, Universitas Gadjah Mada  
Bulaksumur, Yogyakarta 55281

<sup>3</sup>Graduate School of Science Education, Universitas Bengkulu  
Kandang Limun, Bengkulu 38371  
e-mail: esa.apriaskar@mail.unnes.ac.id

**Abstrak**—Artikel ini mendeskripsikan desain dan implementasi algoritma *behavior-based robotic* (BBR) pada *wheeled mobile robot* (WMR) Pioneer P3-DX dalam misi menjelajahi maze menggunakan aplikasi simulasi V-REP. Robot harus melacak dan mencari target yang diletakkan secara acak pada sebuah labirin. Setelah sukses mencapai tujuan, robot berjalan kembali menuju posisi awal (*home position*) menggunakan jalur terdekat. Sistem navigasi robot memanfaatkan algoritma BBR untuk mencapai target menggunakan behavior modules yang bekerja secara simultan untuk mendapatkan trajektori robot yang diinginkan. *Behavior* paling dasar yang sangat mampu untuk dibangun dalam sebuah sistem robot adalah *wall-following behavior*. Untuk membuat robot dapat mengikuti dinding dalam kondisi yang aman, lancar, dan responsif, pengendali *proportional-integral-derivative* (PID) digunakan. Kontroler PID bekerja dengan memanfaatkan pembacaan enam belas buah sensor proximity yang terpasang dalam robot Pioneer P3-DX terhadap jarak dinding yang diharapkan ketika robot tersebut menjelajahi labirin. Untuk memastikan sistem yang didesain berjalan dengan baik, dilakukan beberapa pengujian yang meliputi pengujian BBR dan pengujian pengendali PID. Pengujian BBR menunjukkan bahwa sistem mampu memilih jalur terdekat saat kembali menuju posisi awal. Pengujian pengendali PID menghasilkan pergerakan robot dengan *maximum deviation* dan *settling time* masing-masing berkisar 0,013 m dan 30 detik.

**Kata kunci:** *aplikasi simulasi V-REP, robot pioneer P3-DX, behavior-based robotic, maze solving*

**Abstract**— This article describes the design and implementation of behavior-based robotic (BBR) algorithm on a wheeled mobile robot (WMR) Pioneer P3-DX in a maze exploration mission using V-REP simulator. This robot must trace and search for targets placed randomly on a labyrinth. After successfully meeting the objective, robot runs back to home position using the nearest path. Robot navigation system applies BBR algorithm to reach the target using behavior modules which work simultaneously to obtain the desired robot's trajectory. The most fundamental behavior which is highly affordable to build on the robot system is a wall-following behavior. To make the robot could follow the wall in a safe, smooth and responsive condition, proportional-integral-derivative (PID) controller is applied. PID controller runs by utilizing the reading of sixteen proximity sensors carried on Pioneer P3-DX robot toward the expected wall distance while the robot is exploring the labyrinth. To ensure the designed system works properly, several tests were conducted, including BBR test and PID controller test. BBR test shows that the system can choose the shortest track when returning to home position. The PID controller test produces robot movement with maximum deviation and settling time for about 0.013 m and 30 seconds, respectively.

**Keywords:** *V-REP simulator, pioneer P3-DX robot, behavior-based robotic, maze solving*

Copyright © 2020 Jurnal Rekayasa Elektroika. All right reserved

## I. INTRODUCTION

Autonomous mobile robot (AMR) is a mobile robot capable of performing tasks in an environment without ongoing guidance from humans. In practice, AMR can be defined as a combination of electronic and mechanical devices such as motors, cables, and electronic items that can perform human-like tasks such as moving, warning

against danger, entertaining, and much more. The limits of the AMR come from the design of the robot itself [1].

The primary abilities every AMR must possess are moving, autonomous-drive, and intelligence. Moving means being able to perform basic motions from one place to another with actuators. The autonomous-drive means being able to complete the task without literal human assistance. The intelligence is demanded to carry out this

autonomous-drive [2]. Such intelligence must be able to recognize the environment in its path, adjust it, and use it to make decisions at each subsequent step to reach the goal. This ability involves various components such as sensors, actuators, and devices for decision making.

A traditional way to design robot motion controllers is by using serial processing. However, the behavior-based-robotic (BBR) approach states that intelligence is the result of the interaction between a set of asynchronous behaviors and environments where behavior is a reaction to stimuli [3]. Some examples of this behavior are exploration behavior, goal-oriented behavior, protective behavior, path-following behavior, postural behavior, cooperative or social behavior, and perceptual behavior.

Complicated or risky research requires a simulation system. Research by exploring the planet, integration of actuation, sensing, and making a robust robotics system is hampering without a simulator. In this case, one of the multipurpose robot simulators has been available on a practical scale called the virtual robot experiment platform (V-REP). V-REP is a portable and flexible simulation software that allows the incorporation of various robots and control techniques. This simulator has benefits for its ease to be accessed by the public. It is also helpful in reducing complexity in the world of robot design [4].

BBR-based AMR motion control has widely applied to nonholonomic wheeled robots [5] - [6]. Research conducted by Balch et al. [7] is about the behavior of formation in a multi-robot that is integrated with other navigation behaviors to allow the robot team to reach the goal of navigation and avoid danger simultaneously. Rusu et al. [8] conducted a study of neuro-fuzzy controls for sensor-based mobile robot navigation for indoor conditions by building a hierarchy of robotic behavior.

This study applies BBR to the completion of the Pioneer P3-DX robot in finding the shortest route in the maze using the V-REP simulator. Since the Pioneer P3-DX robot's navigation security is essential in making AMR complete the mission, therefore, to cause robotic behavior follows the wall in a safe, smooth, and responsive condition, proportional-integral-derivative (PID) controller is applied to the system. The controller performs utilizing sixteen proximity sensors carried in Pioneer 3-DX robot to maintain the expected wall distance while the robot is exploring the labyrinth.

This paper organized as follows. Part II proposes the methodology used in this paper. Part III explains the design of the BBR algorithm on Pioneer P3-DX robots. Section IV presents a system simulation and analysis procedure that shows the benefits of the work, while Part V highlights the conclusion of this study.

## II. METHODS

### A. Behavior-Based Robotic (BBR)

The general approach to building a robot control system is to describe every problem into a series of functional

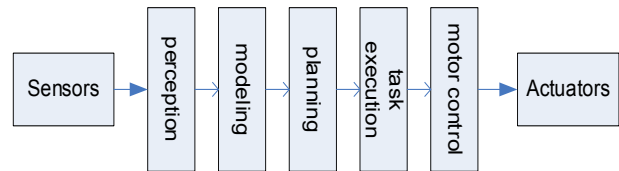


Figure 1. Traditional decomposition techniques for mobile robot control systems into functional units

units, as shown in Fig 1.

In building the robot, BBR structure proposed by Rodney Brooks [9] described the problem of the robot control system in accordance with the external desired manifestations. It does not base on internal operations of the robot control system, as had been conducted by several previous researchers. Therefore, Brooks defined some competency levels in independent mobile robots. Competency level is an informal specification of a group of behaviors that the robot desires to serve in all possible environments. A higher competent level indicates a more specific group of behaviors.

Each competency level incorporates subgroups of previous competency levels. Since the competency level defines a valid behavioral group, it can consider that a higher level provides additional limitations in that behavioral group. Rodney Brooks started by building a control system for a robot that applied zero-level competency. The error was corrected thoroughly. Brooks had never changed this system and called it the zero-level control system. Moreover, he built another control layer called the first-level control system. This level can test data from the zero-level and provide data into the internal interface of the zero-level. It can also suppress standard data flowing. This layer, with an addition of the zero-level, carries out the first-level competency. The zero-layer remains to operate even the layer above sometimes interferes with the data flow. The same process is repeated to get a higher competency level depicted in Fig. 2.

### B. V-REP, Pioneer P3-DX

V-REP is a 3-dimensional (3D) robot simulator with a distributed control architecture: control programs (or scripts) can be directly attached to the object and run together with various threaded types. Thus, V-REP is very flexible and ideal for multi-robot applications and allows users to model robotic systems in the same way as they are. Examples of robot features in V-REP software are like the Humanoid Robot NAO, Wheeled Robot Pioneer P3DX, Epuck, Arm Manipulator Robot [10].

Pioneer P3-DX is a type of two-wheeled mobile robot with a differential steering configuration, which means that the rotating direction of the robot determined by two variables, namely the rotating speed of the left wheel and the right wheel. Pioneer P3-DX, as shown in Fig 3, is the most popular mobile robot platform in the world. The flexibility, reliability, and durability of Pioneer P3-DX have made it a reference platform for robotics research.

Based on Fig. 4, differential steering movement of the

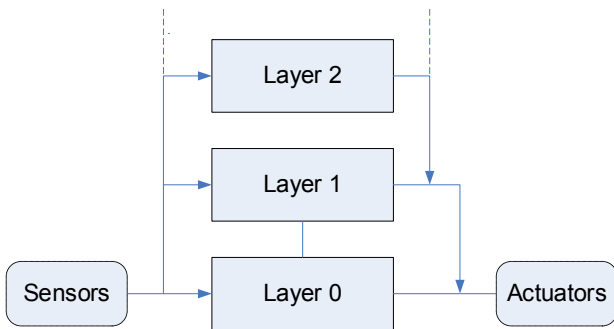


Figure 2. Repetitive processes for the higher competency level

robot is explained as follows:

- Position (a). If both motors twist in the same direction and velocity, the robot will go straight forward. If the two motors do not rotate at the same speed, then the robot will move forward unison; if the right actuator is faster than the left one, then the robot will move forward on the left side, and vice versa.
- Position (b). The left and right motors rotate in the opposite direction. In this case, the robot will turn left/ counterclockwise with the center of rotation on the black point (pivot point). For this maneuver, the two motors must rotate at the same speed so that the pivot point stays in the center.
- Position (c). It has the same idea with position (b) for the opposite direction and the pivot point, but this maneuver addressed to clockwise rotation.
- Position (d). If both motors rotate with the same direction and speed, the robot will go backward straight. If the two motors do not rotate at the same rate, the robot will move back and forth. If the right actuator is faster than the left one, then the robot will move back and forth left, as well as vice versa.

### III. SYSTEM DESIGN

#### A. System Specification

The system defined in this paper is the Pioneer P3-DX robot navigation control using BBR approach with specifications as follows.

- The wheeled robot uses a differential steering method that is capable of smooth and safe maneuvering in the labyrinth.
- It equipped with sensors to support the left or right search algorithm in the maze.
- The robot can trace the left or right side of the maze wall.
- The robot can complete the maze problem.



Figure 3. Pioneer P3-DX

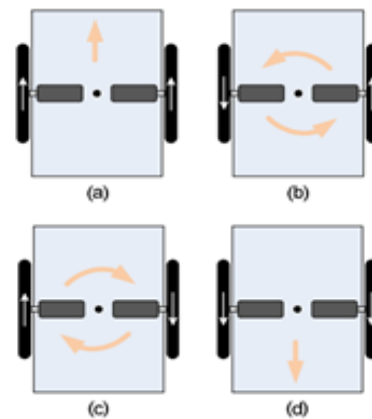


Figure 4. Differential steering movement of Pioneer P3-DX robot

#### B. Simulation Design using V-REP

In designing the simulation, the authors provide a complicated road network as a maze. There are two types of mazes commonly used in robotics, viz wall maze and line maze. The wall maze is generally known as the labyrinth with walls but without roofs. The problem that the robot needs to solve is finding the shortest path.

Maze mapping is an algorithm used to solve a maze, which is describing maps of the labyrinth [11]. Maze mapping in various sources explained as path mapping with the basic concept based on a left /right-hand rule. It used commonly at wall follower robots in an international event called micro-mouse [12]. This algorithm is the basic one. It has the option to drive following the left or right wall in the process of mapping the maze. If the generated map is applied, the robot can return to the starting position through its shortest path. It is also able to repeat through the shortest path from the start point to the target.

This paper explains how the BBR approach, as an essential foundation, can solve the maze problem demonstrated in Fig 5. The required actions to manage the robot's navigation in the maze are listed below.

- Safe navigation (not touching the wall).
- Follow the wall.
- Able to identify the intersection.
- Decide the direction to choose from an intersection.

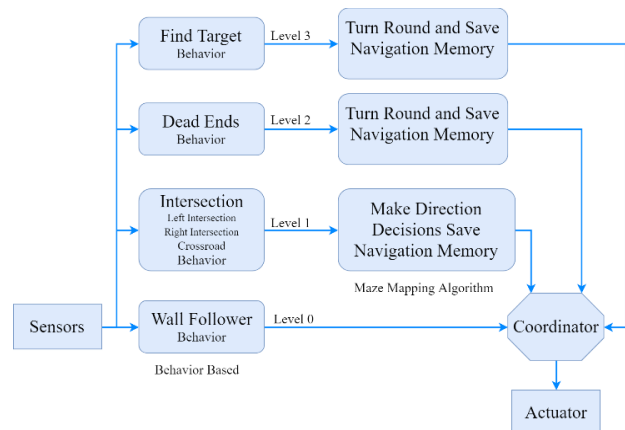


Figure 5. Flowchart of behavior-based robotic algorithm

- e. Take the direction of the intersection according to the left or right-search mode.
- f. Recognize dead ends.
- g. Recognize the target.

To connect the four behaviors (layer-0 to layer-3), we added an arbitration coordinator to determine which layer is taking control of the actuator and determine the movement of the robot. It also manages whether intersection behavior or wall-following behavior from the previous level who takes control of the robot's movement when it encounters an intersection. This arbitration coordinator serves as a competitive coordinator, decision making on each behavior can be arranged through the competency level, and between fellow behavior cannot influence each other. The results of the competitive coordinator's design of several behaviors can be seen in Fig. 6.

Safe navigation refers to the ability of a robot to be able to move inside a maze safely. Safe here interpreted as not hitting or touching the wall. That is a fundamental requirement for other actions because it is directly related to the safety of the robot itself. Whatever other behaviors it has, safety is always the first prerequisite for robots. Next, so that the robot can continuously improve its position in the labyrinth, the ability to follow the wall is inevitable. With only walls inside the maze, the best guide for robot movements is the wall itself because the walls do not change and have a fixed position, and there are many available.

After the robot can move safely and improve its position, the next thing that is needed by the robot to be able to find the path of the recording towards the target is the ability to recognize the intersection. The intersection is a point when a track split into two or three. The inability to identify intersections will make the robot's efforts to reach the target become very difficult. The illustration of right-hand three-way intersection explained in Fig. 7. So, the robot must not only recognize the intersection but also be able to decide the direction to be taken and save the navigation memory into unique codes. Those particular codes lead to the shortest path. This solving maze algorithm named with the maze mapping algorithm, this algorithm uses the left or right search method only. Fig. 8 explains the illustrations in each behavior that starts

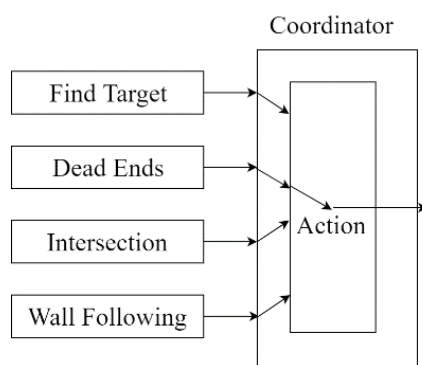


Figure 6. Design of the arbitration coordinator

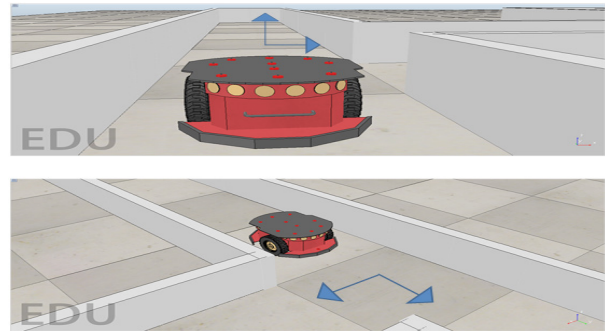


Figure 7. Illustration of right-hand rule in an intersection

from the most basic behavior, namely the wall-following behavior located in layer 0.

#### IV. RESULT AND DISCUSSION

##### A. Behavior-Based Robotic Simulation

The BBR algorithm in this study is simulated using V-REP with Python programming, as presented in Fig. 9. The maze map used and the appearance of the map maze on V-REP are depicted in Fig. 10. There are several letters in the map which become pointers for the position of the robot. The position of the robot decides different strategies that the robot may perform, as explained in Table 1.

Table 1. Behavior test of left-search method in a maze and the solution

Position	Behavior and the maze strategy	Memory
A	The robot departs from home position to the target using the left-search method.	-
B	At this position, the front and right proximity sensors detect the behavior of the "right 3-way intersection," then the robot moves straight and saves the navigation memory with the code "S".	S
C	At this position, the front, left, and right proximity sensors detect a "dead end" behavior, so the robot turns around and saves the navigation memory with the code "U".	U
D	At this position, the front and left proximity sensors detect the "left 3-way intersection" behavior, then the robot turns left and saves the navigation memory with the code "L".	L
E	At this position, the front, left and right proximity sensors detect a "4-way intersection" behavior, then the robot turns left and saves the navigation memory with the code "L".	L
F	Same with position C.	U
G	Same with position E.	L
H	Same with position C and F.	U
I	Same with position E and G.	L
J	Same with position D.	L
K	The robot stops because it has found the target.	-



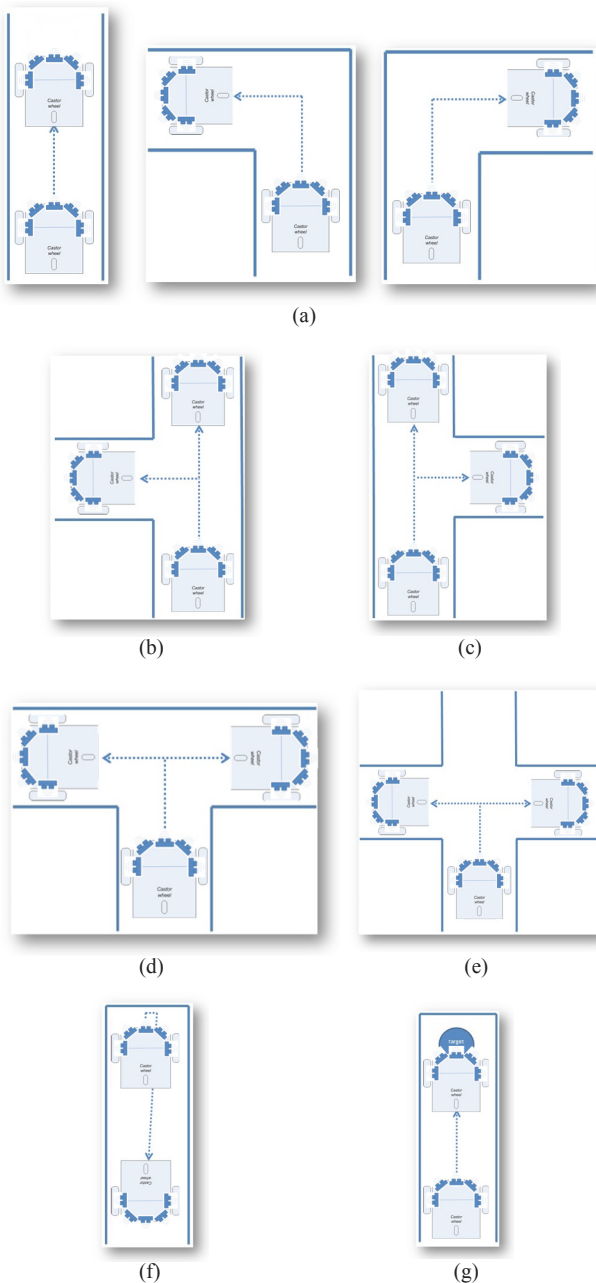


Figure 8. (a) illustration of the left wall-following behavior when using left-search and the right wall-following behavior when right-search mode Illustrations for each behavior, (b) illustration of the left 3-way intersection behavior: if using left-search mode then turn left, but if using right-search mode then straight forward, (c) illustration of the right 3-way intersection behavior: if using right-search mode then turn right, but if using left-search mode then straight forward, (d) illustration of the T-intersection behavior: if using left-search mode then turn left, but if using right-search mode then turn right, (e) illustration of the 4-way intersection behavior: if using left-search mode then turn left, but if using right-search mode then turn right, (f) illustration of the dead-end behavior then the robot turns around, (g) illustration of finding target behavior

**B. Behavior-Based Robotic Analysis**

This analysis is carried out by placing a target (written ‘Finish’) on the labyrinth corridor, then the robot runs from the origin position (written ‘Start’), uses left or right-search method towards the target. During the robot runs, the sensor stimulates various behaviors. The architecture

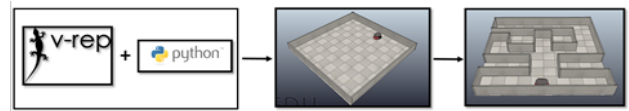


Figure 9. Diagram block of the simulation

of the BBR algorithm plays a vital role in seeking the target. When the robot detects an intersection, which lets the intersection-behavior active, the coordinator prioritizes this behavior in the action of actuator movement. For more details on the issue of architectural BBR testing on Pioneer P3-DX robot, consider the illustration of Pioneer P3-DX robot testing uses the left-search method and the unique codes, described in Fig. 11 and Table 1.

From the exploration of Pioneer P3-DX robots using the left-hand rule on the maze track, a navigation memory record is formed with the following code and simplified to be “RRL” as seen in Fig. 12. It means that each intersection starts from the robot’s direction and navigates to the right (R) for the first intersection, right (R) for the second intersection, and left (L) for the third intersection.

**C. Behavior-Based Robotic Test**

The wall-following behavior runs based on the proximity sensor readings on Pioneer P3-DX robots.



Figure 10. The representation of the maze map on V-REP

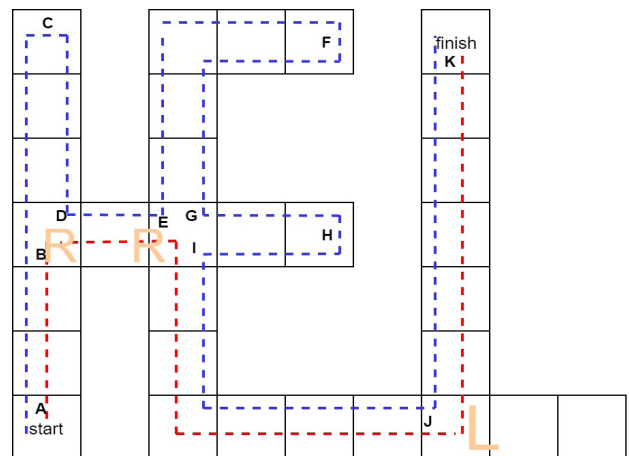


Figure 11. Illustration of Pioneer P3-DX testing uses the left-search method

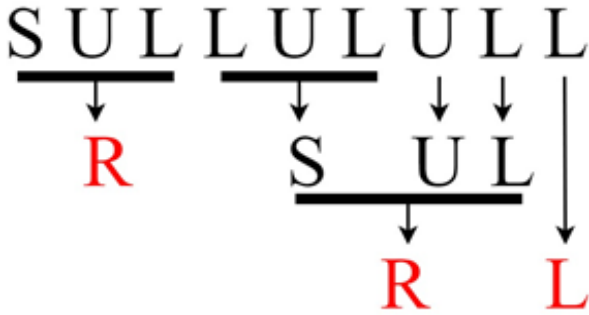


Figure 12. The simplicity of the recorded navigation memory

Table 2. Detection data using proximity sensors employed in pioneer P3-DX robot

No	Conditions for Objects	Proximity sensor readings (m)
1	Detected	0,041954044
2	Detected	0,141878769
3	Detected	0,216879264
4	Detected	0,316879392
5	Detected	0,416879505
6	Detected	0,516879141
7	Detected	0,591878951
8	Not Detected	0,656990032

The data on reading the wall are shown in Table 2. The characteristics of proximity sensor readings in Pioneer P3-DX robots is that when the gap between the sensor and the wall is too close, the reading value is zero or even does not detect. Similarly, if the distance of the sensor with the wall is too far away until it is unable to discover, the reading value of the sensor is also zero. The minimum sensor distance reading tolerance detected is 0.04 meters. While the maximum distance when the sensor detects a wall is 0.59 meters, more than that, the proximity sensor does not recognize the presence of walls or objects.

D. Driving Time of The Robot

Table 3 presents data about the time taken by Pioneer P3-DX robots from A to K with the left-search method, as described in Fig. 11. The distance of A-B, B-C, and C-D could take approximately twelve seconds. Whereas the interval between D-E, G-H, and H-I could spend around seven to ten seconds. For longer distances such as distance E-F, F-G, and I-J can be reached in twenty-one to twenty-nine seconds. The total time needed by the robot from A to K is 161 seconds using the left search method.

E. PID Control Test

The PID controller implemented in this paper works based on the digital PID control equation with an independent structure written in (1). First, two variables created in the form of errors and last error to save the current and previous value about the gap between desired

Table 3. The driving time of Pioneer P3-DX robot

Distance	Start	Stop	Time (s)
A-B	00.44	00.56	00.12
B-C	00.56	01.07	00.11
C-D	01.15	01.27	00.12
D-E	01.30	01.40	00.10
E-0	01.44	01.56	00.12 00.21
0-F	02.08	02.17	00.09
F-0	02.25	02.37	00.12 00.25
0-G	02.43	02.56	00.13
G-H	03.00	03.07	00.07
H-I	03.16	03.24	00.08
I-0	03.28	03.41	00.13 00.29
0-J	03.45	04.01	00.16
J-K	04.04	04.30	00.26

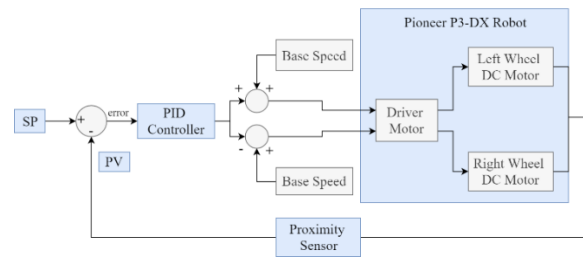


Figure 13. Block diagram of the PID control test

and the present value of the sensor. Then, the calculation of the PID control actions utilizes those two variables. Once the program is repeating, the current reading of the sensor will update the error data and store the previous error data into the last error. The output of the PID controller is a speed that can be positive or negative. Positive and negative indicate the direction of rotation of the motor. If it is positive, then it will rotate forward; conversely, if it is negative, then the motor will reverse. This value will also be added and subtracted first with the base speed for each actuator and added to the offset of the speed of each actuator, as described in Fig. 13.

$$u_{(k)} = K_p e_k + K_i T \sum_0^k e_k + \frac{1}{T} K_d (e_k - e_{k-1}) \tag{1}$$

PID control test conducted by varying one parameter alternately while the other two parameters are fixed at first and done in steps. This test is carried out on the test track presented in Fig. 14.

The tuning method used in the PID control test is a trial and error experiment with regards to the characteristics of the PID control [13]. The first step is using only



Figure 14. Track for PID control test

Table 4. Proportional control test

Kp	Ki	Kd	Pioneer P3-DX Condition
0.5	0	0	The robot's response is prolonged and less stable
2	0	0	The robot's rise time increases, but still less stable
4	0	0	The robot is unstable and oscillating

Table 5. Proportional and derivative control test

Kp	Ki	Kd	Pioneer P3-DX Condition
2	0	3	The robot's response is more stable, and oscillations begin to decrease by 30%
2	0	5	The robot's response is more stable, faster and oscillation is reduced by 50%
2	0	7	The robot's response is too fast and returns to higher overshoot in the beginning

proportional control, ignore the integrative (Ki), and the derivative parameter (Kd) by giving zero values as in Table 4. When the constant of proportional component (Kp) given 0.5, the response of the Pioneer P3-DX robot looks very slow and less stable, then the Kp value needs to be raised again to get a faster response. Then the Kp value is given two, which results in a better response, but still

Table 6. Proportional, Integral and Derivative Control Test

Kp	Ki	Kd	Pioneer P3-DX Condition
2	0.01	5	The robot's response is fast and stable
2	0.05	5	The robot's response is fast and more stable
2	0.5	5	The robot's response is too fast and oscillation increases

unstable. However, when Kp assigned a value of four, the Pioneer P3-DX robot became unstable. An extremely high oscillation may cause the Pioneer P3-DX robot to hit the wall pictured in Fig. 15(a).

From Table 4, proportional control testing has not been able to make the Pioneer P3-DX robot move fast and stable with regards to the desired trajectory. The addition of Kd is conducted at the next stage to reduce the oscillation of the response. Table 5 is proportional control test (Kp = 2) with addition of derivative control. From Fig. 15(b), it is clear that the characteristics of Kd can reduce oscillations caused by proportional control and make the movement of Pioneer P3-DX robots more stable.

Furthermore, if the robot has been stable enough, the integrative controller can be just an optional, giving improper integral parameter can make the robot system perform more oscillation, presented in Table 6 and Fig.

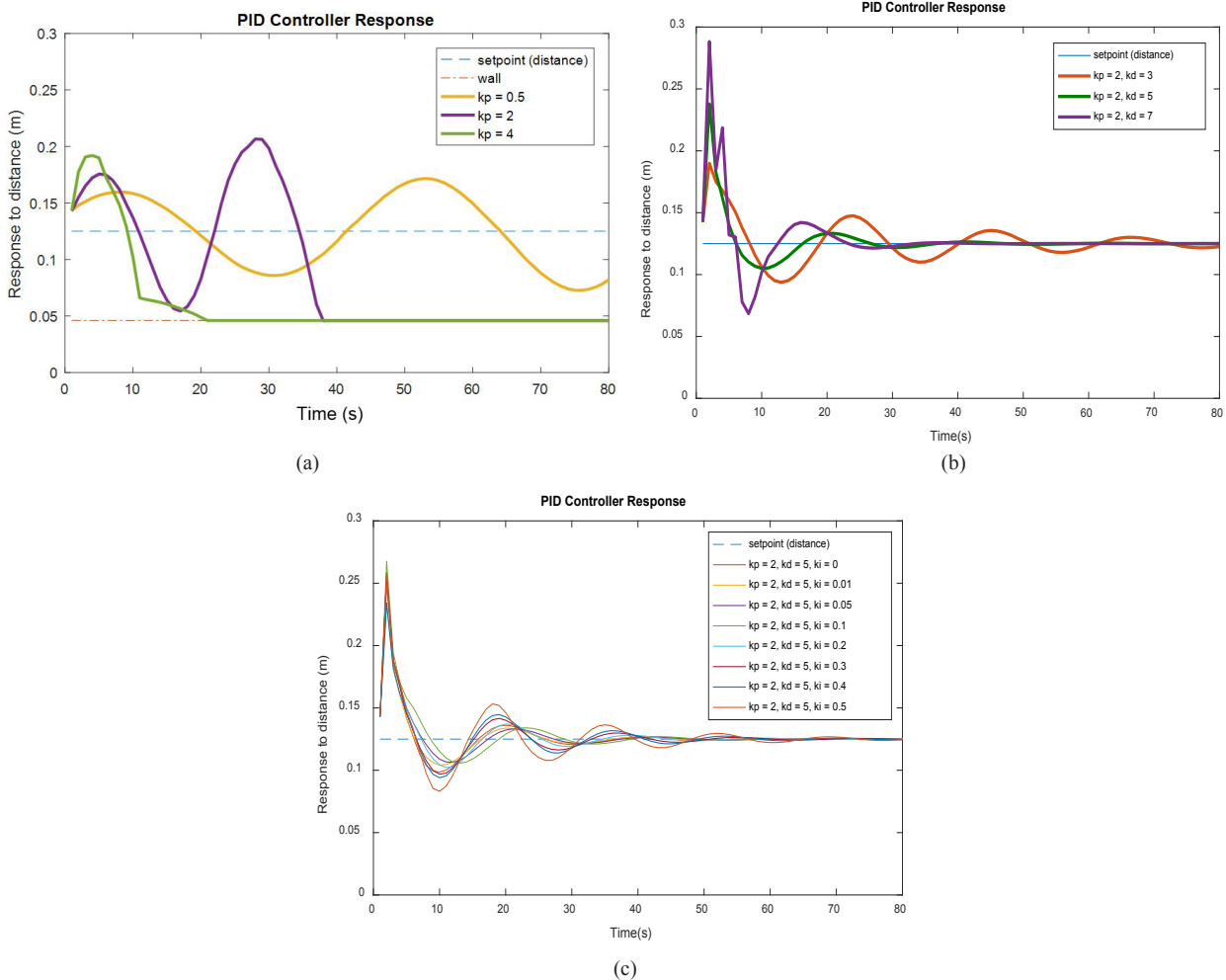


Figure 15. The response resulted in tuning (a) proportional parameter, (b) derivative parameter, and (c) integral parameter



15(c). When  $K_i$  has the value of 0.01 through 0.4, the change is not too significant. Though, when the  $K_i$  value increases to 0.5, oscillation increases even higher than without integral control.

From the results of proportional, integral and derivative control test, it indicates that the robot's response will be better if the use of integral part is proper. When the use of integral is too excessive, it can cause the Pioneer P3-DX robot to oscillate and become more oscillatory as in Fig. 15(c). Chosen parameters to give the most satisfactory response are  $K_p = 2$ ,  $K_i = 0.05$  and  $K_d = 5$ . It has the best performance in maximum deviation and settling time for about 0.013 m and 30 seconds, respectively. To improve the performance of the controlled system, it is considered to use a tuning method for PID controller, like Ziegler-nichols [14]. It is also possible by upgrading the controller into a more advanced PID structure as described in [15] or combined with optimization techniques to precisely determine the controller parameters, as mentioned in [16], [17].

## V. CONCLUSION

From the observations, testing, and analysis, we can draw several conclusions. This paper presents the design and implementation of the behavior-based robotic (BBR) algorithm on a wheeled mobile robot (WMR) Pioneer P3-DX to explore a maze in a V-REP simulator environment. The results obtained are that behavior-based control architecture is very concerned with continuous testing at each level of its behavior. It is not a good idea to add a new behavior when previous behavior cannot run properly. In addition, the application of PID control on Pioneer P3-DX robot has been adequate to make the movement of the robot stable and able to make the robot maneuvering safely, smoothly, and responsively. The PID control parameters obtained from the experimental tuning are  $K_p = 2$ ,  $K_i = 0.05$  and  $K_d = 5$ , which resulted the best performance in maximum deviation and settling time. For future work, the controller can be considered to use structured tuning method. It can also be upgraded into a more advanced PID structure or combined with optimization techniques to determine the controller parameters.

## REFERENCES

- [1] H. M. La, R. S. Lim, B. B. Basily, N. Gucunski, Jingang Yi, Ali Maher, F. A. Romero, and H. Parvardeh, "Mechatronic systems design for an autonomous robotic system for high-efficiency bridge deck inspection and evaluation," *IEEE/ASME Transactions on Mechatronics*, vol. 18(6), pp. 1655-1664, December 2013.
- [2] T. Das and I. N. Kar, "Design and implementation of an adaptive fuzzy logic-based controller for wheeled mobile robots," *IEEE Transactions on Control Systems Technology*, vol. 14(3), pp. 501-510, May 2006.
- [3] Fahmizal, Ting-Shuo Chen, Shou-Wei Chi, and Chung-Hsien Kuo. "Fuzzy controller based subsumption behavior architecture for autonomous robotic wheelchair," in 2013 International Conference on Advanced Robotics and Intelligent Systems, May 2013, pp. 158-163.
- [4] E. Rohmer, S. P. Singh, and M. Freese, "V-REP: A versatile and scalable robot simulation framework," In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, November 2013, pp. 1321-1326.
- [5] Yan Meng, Xiyu Liu, and Zhen Liang. "A Robust Method for Mobile Robot Tracking Object on Behavior-Based Robotics." In *First IEEE International Symposium on Information Technologies and Applications in Education*, November 2007, pp. 468-472.
- [6] D. M. Lyons, R. C. Arkin, Shu Jiang, Tsung-Ming Liu, and P. Nirmal, "Performance verification for behavior-based robot missions," *IEEE Transactions on Robotics*, vol. 31(3), pp. 619-636, June 2015.
- [7] T. Balch and R. C. Arkin. "Behavior-based formation control for multirobot teams," *IEEE Transactions on robotics and automation*, vol. 14(6), pp. 926-939, December 1998.
- [8] P. Rusu, E. M. Petriu, T. E. Whalen, A. Cornell, and H. J. W. Spoelder, "Behavior-based neuro-fuzzy controller for mobile robot navigation," *IEEE transactions on instrumentation and measurement*, vol. 52(4), pp. 1335-1340, August 2003.
- [9] R. Brooks, "A robust layered control system for a mobile robot," *IEEE Journal on Robotics and Automation*, vol. 2(1), pp. 14-23, March 1986.
- [10] M. Freese, S. Singh, F. Ozaki, and N. Matsuhira, "Virtual robot experimentation platform V-REP: A Versatile 3D Robot Simulator," In *International Conference on Simulation, Modeling, and Programming for Autonomous Robots*, November 2010, pp. 51-62.
- [11] I. E. Sutherland, "A method for solving arbitrary-wall mazes by computer," *IEEE Transactions on Computers*, vol. C-18(12), pp. 1092-1097, December 1969.
- [12] Hsin-Hsiung Huang, Juing-Huei Su, and Chyi-Shyong Lee, "A contest-oriented project for learning intelligent mobile robots," *IEEE Transactions on Education*, vol. 56(1), pp. 88-97, February 2013.
- [13] Fahmizal, M. Arrofiq, and A. Mayub. "Identifikasi Pemodelan Matematis Robot Wall Following," *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, vol. 7(1), pp. 79-88, February 2018.
- [14] E. Apriaskar, Fahmizal, N. A. Salim, and D. Prastiyanto. "Performance Evaluation of Balancing Bicopter using P, PI, and PID Controller," *Jurnal Teknik Elektro (JTE)*, vol. 11(2), pp. 44-49, December 2019.
- [15] N. Tamami, I. M. Diin, B. Sumantri, and E. Pitowarno. "Robot Keseimbangan Beroda Dua dengan Sistem Kontrol Keseimbangan dan Posisi Menggunakan Metode PID Bertingkat." *Jurnal Rekayasa Elektrika*, vol. 14(3), pp. 181-188, December 2018.
- [16] B. Y. Suprpto and Sariman. "Metode Algoritma Genetika dengan Sistem Fuzzy Logic untuk Penentuan Parameter Pengendali PID," *Jurnal Rekayasa Elektrika*, vol. 10(1), pp. 32-38, April 2012.
- [17] A. Triwiyatno, Sumardi, and E. Apriaskar. "Robust Fuzzy Control Design using Genetic Algorithm Optimization Approach: Case Study of Spark Ignition Engine Torque Control," *Iranian Journal of Fuzzy Systems*, vol. 14(3), pp. 1-13, June 2017.

**Penerbit:**

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Syiah Kuala

Jl. Tgk. Syech Abdurrauf No. 7, Banda Aceh 23111

website: <http://jurnal.unsyiah.ac.id/JRE>

email: [rekayasa.elektrika@unsyiah.net](mailto:rekayasa.elektrika@unsyiah.net)

Telp/Fax: (0651) 7554336

