

Implementation of Pure Pursuit Algorithm for Nonholonomic Mobile Robot using Robot Operating System

Omur Aydogmus and Gullu Boztas

Abstract— In this study, a differential wheeled mobile robot was controlled in real time using pure pursuit algorithm (PPA). The robot was obtained in a simulation environment by using Gazebo simulator which offer the ability to accurately and efficiently simulate various robots in complex indoor/outdoor environments. This simulator was operated with robot operating system (ROS) which allows the use of Python, C++, MATLAB or various programming languages. In this paper, MATLAB/Simulink environment was used to control the robot with communication interface between MATLAB and ROS. Thus, it is possible to study more comprehensively by using multiple the features of MATLAB. The robot was traveled around a 4m x 4m area with random waypoints. The position of the robot was measured with odometer sensor in order to determine the robot's location. The performance of the control algorithm was analyzed by using various information of the robot such as robot velocity, motors speed, the robot position, etc.

Index Terms— Mobile Robot, Pure Pursuit Algorithm, Robot Operating System.

I. INTRODUCTION

THE SCIENTIFIC researches on mobile robots are increasing rapidly day by day. Legged robots, wheeled robots, underwater robots, and aerial robots constitute various sub-branches of mobile robots [1]. The mobile robots are used in medical facilities [2], customer service [3], agriculture [4], warehousing [5], package delivery [6], disaster recovery [7], emergency response [8], entertainment [9], space exploration [10], defense and so on. The mobile robots are used in nonindustrial applications and industrial applications. Wheeled mobile robots are used in some applications because of their many advantages such as rapidity, accuracy, achieving easily repetitive and hard tasks, and so on. There are two main

types of wheeled mobile robot, non-holonomic and holonomic. Non-holonomic wheeled robots have only 2 degrees of freedom (DOF). One of the movements provides linear movement in the x-axis and the other performs the rotational movement around the z-axis on a planar surface. Holonomic wheeled robots have 3 DOF with higher motion capability. Holonomic wheeled robots can travel along x-axis and y-axis with linear movement. In addition, they can rotate around the z-axis [11]. Holonomic wheels are also called as mecanum and omni-wheel. There are some disadvantages of these wheels such as expensive, heavy, slow, and friction [12]. Thus, non-holonomic wheels are widely preferred instead of the holonomic wheels for mobile robot applications. Recently, the interest of the differential drive wheeled mobile robots (DDWMR) are increasingly noticed and used in many applications. DDWMRs have some advantages such as flexible motion capability, simple and robust structure, lower costs [13]. DDWMRs can be obtained as 2-wheel [14], 3-wheel [15], and 4-wheel [16]. All these structures have only two motorized wheels. The other wheels are used as passive castor wheel which can rotate free spin around the own axis. The castor wheels are used in order to obtain balance and stability. DDWMRs with 3-wheel are widely used in many industrial applications. These mobile robots have more easily control with high maneuverability [17]. Therefore, a 3-wheeled DDWMR was used in the current study. This robot is called as TurtleBot3 Burger which is developed in 2017. TurtleBot3 is an open source based software and hardware. This robot is fully supported from robot operating system (ROS) platform. ROS enables the development and implementation of robot control algorithms, both in the simulation environment and in the real environment. Thus, the programs developed in the simulation environment can be used in real experimental studies without changing them [18]. Mobile robots need to control autonomous or manually determined point-to-point movements [19]. Pure pursuit algorithm (PPA) is widely used for path tracking [20]. The PPA has more simple implementation principle with better tracking results compared with other path tracking algorithms [21], [22]. PPA can be performed by using proportional-integral-derivative (PID) algorithm [23], linear quadratic regulator algorithm [24], model predictive control algorithm [25]. The look-ahead point is a critical parameter for PPA to achieve appropriate results.

OMUR AYDOGMUS, is with Department of Mechatronics Engineering of Firat University, Elazig, Turkey,(e-mail: oydogmus@firat.edu.tr).

 <https://orcid.org/0000-0001-8142-1146>

GULLU BOZTAS, is with Department of Electrical Engineering of Firat University, Elazig, Turkey,(e-mail: gboztas@firat.edu.tr).

 <https://orcid.org/0000-0002-1720-1285>

Manuscript received August 16, 2021; accepted September 20, 2021.

DOI: [10.17694/bajece.983350](https://doi.org/10.17694/bajece.983350)

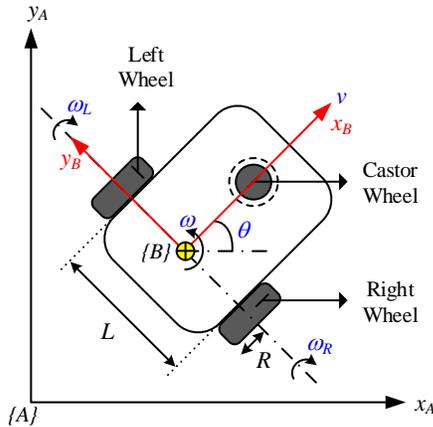


Fig.1. Simplified view of a differential drive wheeled mobile robot.

This study focuses on the look-ahead value to improve the tracking performance of the robot. Therefore, a realistic simulation environment and robot model were obtained by using Gazebo simulator. A TurtleBot3 Burger DDWMR was used for analyzing the PPA. The PPA was developed in MATLAB environment which connected to the ROS nodes. Two PC were used to obtain a realistic result. A PC with Linux operating system was used to simulate both the robot and the environment. This PC ran ROS creates publisher and subscriber nodes over IP address. The PPA was implemented on the other PC with the MATLAB program. Both control and monitoring processes were achieved by connecting to ROS nodes via IP address. The PPA was analyzed for different look-ahead values.

II. DIFFERENTIAL DRIVE WHEELED MOBILE ROBOT

Three wheeled DDWMR was used in this study. The two wheels of the robot are motorized and the other one is the castor wheel which can rotate around its own axis. This robot can only move linearly along the x-axis. It cannot move along the y-axis. In addition, the robot has the ability to rotate around the z-axis. It is possible to control the robot in the xy-plane according to the difference in the angular velocities of the left and right wheels. The simplified robot structure is given in Fig. 1.

The robot motion is represented by linear velocity v (m/s) and angular velocity ω (rad/s). The kinematics equations are derived by using the parameters shown in the Fig. 1. The forward kinematics of the robot is given in Equations (1) and (2). The inverse kinematics of the robot is given in Equations (3) and (4).

$$v = \frac{R}{2}(\omega_R + \omega_L) \quad (1)$$

$$\omega = \frac{R}{L}(\omega_R - \omega_L) \quad (2)$$

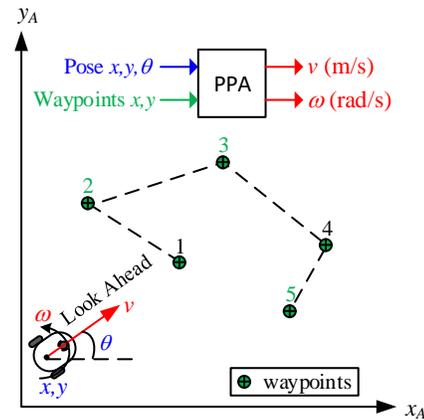


Fig.2. Inputs/outputs of PPA and other parameters on coordinate frame.

$$\omega_L = \frac{1}{R} \left(v - \frac{\omega L}{2} \right) \quad (3)$$

$$\omega_R = \frac{1}{R} \left(v + \frac{\omega L}{2} \right) \quad (4)$$

where R is the wheel radius of the left/right wheels, L is the wheelbase of the robot, ω_L and ω_R are the angular velocity of left and right wheel, respectively. As shown in the forward kinematics equations, the robot can be controlled by adjusting the velocities of the left and right wheels. If both wheel velocities are adjusted equal with opposite direction to each other, the robot can be moved either forward or backward in a straight line along the x-axis. The wheel coordinate systems are not equal. Therefore, the direction of the rotating wheels must be unequal for linear motion. If the robot is to be rotated about the z-axis in a positive direction, then the left wheel should be chosen positive with respect to its own axis and the right wheel should be chosen as negative with respect to its own axis.

III. PURE PURSUIT ALGORITHM

The PPA is the popular tracking algorithm for mobile robot. This algorithm calculates the linear velocity v and angular velocity ω of the robot by using the current pose of the robot and the set of waypoint. The PPA has two inputs and two outputs as shown in Fig. 2. The pose consists of the robot position represented on the xy-plane. The pose of the robot represents the x and y points relative to the A coordinate frame and its angle relative to the x-axis.

The look-ahead distance affects the performance of the algorithm. If a small look-ahead distance is taken, it can improve the robot's path tracking. However, the system instability may occur if look-ahead is not well tuned. Therefore, this value must be determined appropriately. The effect of look-ahead is analyzed for different values as shown in Fig. 3. A smoother path tracking can be obtained as the value of look-ahead increases. However, a course change occurs before reaching the waypoints. If the look-ahead distance decreases, the robot changes course after reaching the

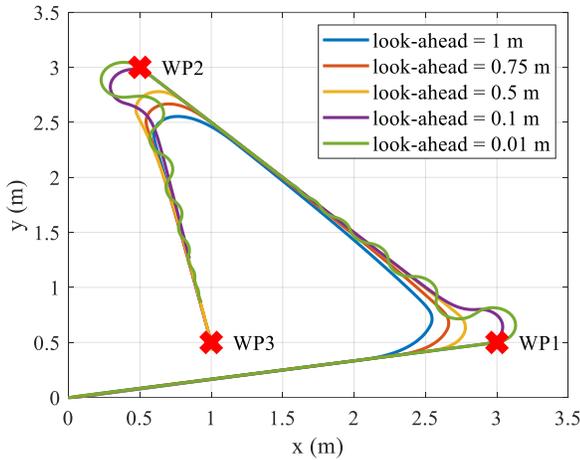


Fig.3. Effect of different look-ahead values for path tracking.

waypoints. This state causes undesirable oscillations on the robot path as shown in Fig.3. In the study, only kinematics equations were used without considering the dynamics conditions. The path followed by the robot has both a higher oscillation and a longer distance, as indicated by the green-line when the look-ahead value is set to 0.01 m. For this reason, it is necessary to set the most appropriate value for the PPA.

IV. IMPLEMENTATION OF PPA FOR DDWMR

In this study, TurtleBot3-Burger was used to implement the PPA. This robot provides open-source features for program developers. The technical specification of the robot are as follows; Maximum translational velocity: 0.22 m/s, Max. rotational velocity: 162.72°/s, Max. payload: 15 kg, Dimensions: 138 × 178 × 192 mm, Weight (+ SBC + battery + sensors): 995 g, Max. traversable step: 10 mm. The mechanical properties of the robot are provided by Gazebo simulator environment, as shown in Fig.4. The Gazebo simulator is an open-source 3D simulation environment to create a robot having realistic dynamics and realistic rendering view in complex indoor and outdoor environments. This simulator supports some physics engine such as ODE, Bullet, Simbody, DART, etc. Thus, the developed robots and the algorithms can be tested with a realistic scenario. In addition, the simulator provides sensors and noise data in real-time. All actuators and sensors can be connected through TCP/IP as a remote interface service.

The robot simulation environment was launched via ROS-Noetic. ROS was created as master server to obtain connection between robot and control algorithm. The structure of the connection between ROS and control PC are shown in Fig. 5. ROS server was operated in a PC which powered by Linux operating system. The Gazebo simulator was also installed the same PC. The control algorithm was developed by using MATLAB environment. The developed control algorithm was communicated with ROS nodes as shown in Fig. 5. ROS has a communication interface that enables send and receive robot data between any other program languages or environment. In the real application, a real robot is used instead of a Linux-installed PC. Thus, the developed algorithms can be tested both simulation environment and real environment without any changes. MATLAB can also connect to a real ROS

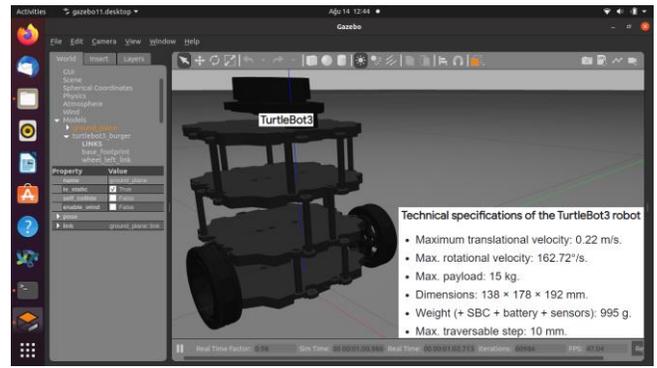


Fig.4. TurtleBot3-Burger by Gazebo simulator environment.

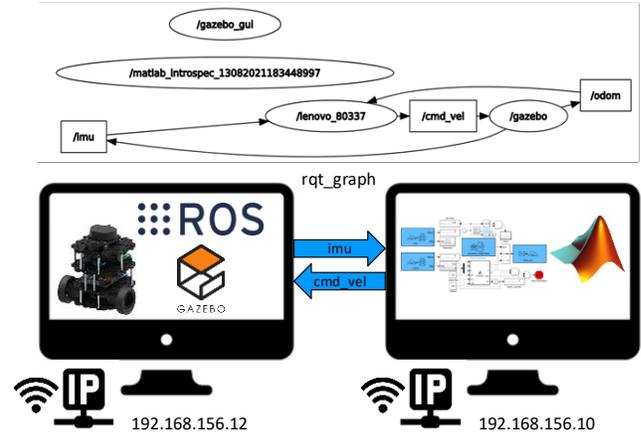


Fig.5. Block diagram of connection between ROS-Gazebo and MATLAB.

network to access all sensor and actuator information which are provided from the simulator. The data of inertial measurement units (IMU) and odometer (ODOM) are generated from Gazebo simulator in real-time. The IMU and ODOM are published on ROS topics as shown labeled as “rqt_graph” in Fig. 5. The published ODOM data includes the information of robot position and orientation on the xy-plane. These variables are used in order to control robot.

The PPA was used to control the robot. The PPA requires the robot’s *position.x (m)*, *position.y (m)*, and *orientation.θ (rad)*. MATLAB subscribed the ODOM through the ROS topics. The PPA uses two geometrical equations to calculate the distance between robot pose and waypoint, and the required angle of the course. The equations for PPA calculations are given in Equation (5) and (6).

$$d = \sqrt{(x_{ps} - x_{wp})^2 + (y_{ps} - y_{wp})^2} \quad (5)$$

$$c = \text{atan2}((y_{wp} - y_{ps}), (x_{wp} - x_{ps})) \quad (6)$$

where *d* is the distance between robot pose and the reference waypoint, *c* is the course of path. *x_{ps}* any *y_{ps}* are the robot positions on the xy-plane. *x_{wp}* any *y_{wp}* are the waypoints on the xy-plane.

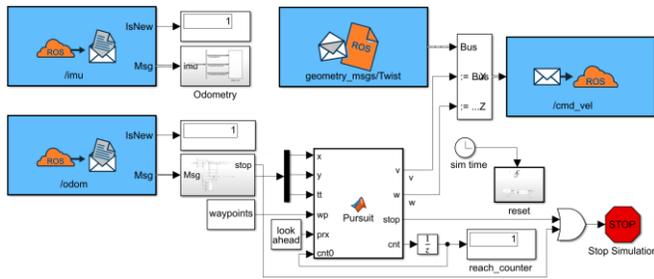


Fig.6. Block diagrams of MATLAB environment for PPA.

The course was used to calculate the reference orientation movement of the robot. The difference between course value and the robot actual orientation value is used in the PPA. The sign of differential value determines the rotation direction of the robot clock-wise or counter clock-wise. All movements were limited by the robot's specifications. The linear velocity and the angular velocity were limited as 0.22 m/s and ± 2.84 rad/s, respectively.

The developed algorithm in MATLAB environment is given in Fig.6. The blue colored boxes are related to ROS. Some of them are subscribers and publishers. "/imu" and "/odom" are subscribers to give data from the ROS topics. "/cmd_vel" is a standard movement command of ROS. This command was published from the MATLAB as a geometry messages for ROS. The algorithm was implemented with three waypoints as WP1(3;0.5), WP1(0.5;3), and WP1(1;0.5). In this study, both the kinematic and dynamic states of the robot are considered. Therefore, the data measured from the robot is very similar to the real robot data.

The results are obtained for five different look-ahead values as shown in the Fig. 7a. The results show that the robot had not an oscillation depending on the look-ahead values. However, the robot had oscillation with small look-ahead values in the previous study as shown in Fig. 3. In the last study, the previous oscillations did not occur because the robot was considered in dynamic states. It has been determined that the travel times of the robot are quite different from each other. The travel times of the robot were calculated as 29.7s, 35.6s, 40.6s, 49.2s, and 55.9s for 1 m, 0.75 m, 0.5 m, 0.1 m, and 0.01 m look-ahead, respectively. The reason for the difference in travel times is due to the fact that the robot reduces its speed considerably when it approaches the reference waypoint in PPA control. At the lowest look-ahead value, it is seen in Fig.7b (green-line) that the robot reduces its speed to almost zero to approach the reference point. However, the linear velocity was not decreased as shown Fig. 7b (blue/red-line) when the look-ahead value was increased. The angular velocity of the robot was not affected the look-ahead value as shown in Fig.7c. The occurrence of rotational movements in different time intervals is due to the different access times to the reference points.

V. CONCLUSION

This study presented an implementation of PPA control for a DDWMMR. First, a simplified robot model was used in order to show the effect of the look-ahead distance value for PPA. The

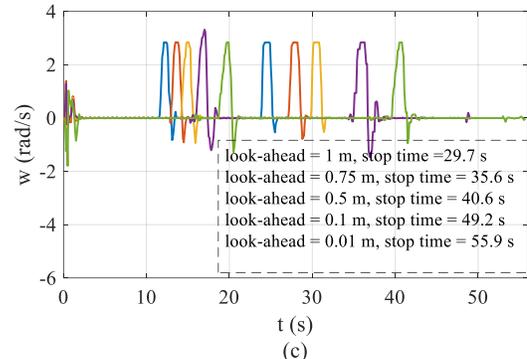
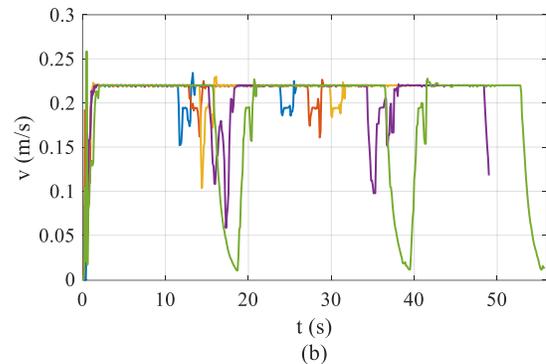
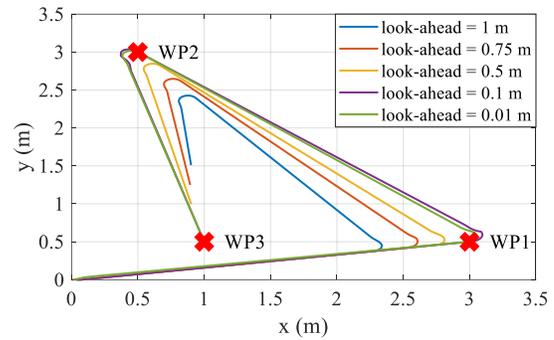


Fig.7. Effects of different look-ahead values for TurtleBot3-Burger; a) Path tracking, b) Linear velocities of robot, c) Angular velocities of robot.

results show that the small look-ahead values had undesirable oscillation on the robot path. Second, a detailed robot model was used by using ROS with Gazebo simulator. A TurtleBot3-Burger was used in the simulator environment to obtain realistic results. The robot was controlled by MATLAB simulation environment which is connected to ROS environment through TCP/IP. The robot can be controlled via PPA which was developed MATLAB environment. The results in the last study showed that the look-ahead values change the robot's reach to the reference points. In addition, no oscillation was observed in the robot movement. It can be said that the highest look-ahead distance positively affects the travel time of the robot. In this case, the robot changes its course without approaching the reference points.

REFERENCES

- [1] F. Rubio, F. Valero, and C. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 2, p. 1729881419839596, 2019, doi: 10.1177/1729881419839596.

- [2] T. Namba and Y. Yamada, "Risks of Deep Reinforcement Learning Applied to Fall Prevention Assist by Autonomous Mobile Robots in the Hospital," *Big Data Cogn. Comput.*, vol. 2, no. 2, 2018, doi: 10.3390/bdcc2020013.
- [3] G. Fragapane, R. de Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *Eur. J. Oper. Res.*, vol. 294, no. 2, pp. 405–426, 2021, doi: <https://doi.org/10.1016/j.ejor.2021.01.019>.
- [4] X. Gao *et al.*, "Review of Wheeled Mobile Robots' Navigation Problems and Application Prospects in Agriculture," *IEEE Access*, vol. 6, pp. 49248–49268, 2018, doi: 10.1109/ACCESS.2018.2868848.
- [5] S. Tanwar, S. Tyagi, and S. Kumar, "The Role of Internet of Things and Smart Grid for the Development of a Smart City," in *Lecture Notes in Networks and Systems*, vol. 19, Springer, 2018, pp. 23–33.
- [6] A. Bärtschi *et al.*, "Collaborative delivery with energy-constrained mobile robots," *Theor. Comput. Sci.*, vol. 810, pp. 2–14, 2020, doi: <https://doi.org/10.1016/j.tcs.2017.04.018>.
- [7] G. Tuna, V. C. Gungor, and K. Gulez, "An autonomous wireless sensor network deployment system using mobile robots for human existence detection in case of disasters," *Ad Hoc Networks*, vol. 13, pp. 54–68, 2014, doi: <https://doi.org/10.1016/j.adhoc.2012.06.006>.
- [8] H. Fan, V. Hernandez Bennets, E. Schaffernicht, and A. J. Lilienthal, "Towards Gas Discrimination and Mapping in Emergency Response Scenarios Using a Mobile Robot with an Electronic Nose," *Sensors*, vol. 19, no. 3, 2019, doi: 10.3390/s19030685.
- [9] E. Jochum, P. Millar, and D. Nuñez, "Sequence and chance: Design and control methods for entertainment robots," *Rob. Auton. Syst.*, vol. 87, pp. 372–380, 2017, doi: <https://doi.org/10.1016/j.robot.2016.08.019>.
- [10] K. Schilling and C. Jungius, "Mobile robots for planetary exploration," *Control Eng. Pract.*, vol. 4, no. 4, pp. 513–524, 1996, doi: [https://doi.org/10.1016/0967-0661\(96\)00034-2](https://doi.org/10.1016/0967-0661(96)00034-2).
- [11] S. Mellah, G. Graton, E. M. El Adel, M. Ouladsine, and A. Planchais, "Actuator Health State Monitoring amp; Degradation Impact Study on a 4-Mecanum Wheeled Mobile Robot Behaviour," in *2021 29th Mediterranean Conference on Control and Automation (MED)*, 2021, pp. 1076–1081, doi: 10.1109/MED51440.2021.9480231.
- [12] Z. Sun, H. Xie, J. Zheng, Z. Man, and D. He, "Path-following control of Mecanum-wheels omnidirectional mobile robots using nonsingular terminal sliding mode," *Mech. Syst. Signal Process.*, vol. 147, p. 107128, 2021, doi: <https://doi.org/10.1016/j.ymssp.2020.107128>.
- [13] A. Stefek, T. Van Pham, V. Krivanek, and K. L. Pham, "Energy Comparison of Controllers Used for a Differential Drive Wheeled Mobile Robot," *IEEE Access*, vol. 8, pp. 170915–170927, 2020, doi: 10.1109/ACCESS.2020.3023345.
- [14] R. P. M. Chan, K. A. Stol, and C. R. Halkyard, "Review of modelling and control of two-wheeled robots," *Annu. Rev. Control*, vol. 37, no. 1, pp. 89–103, 2013, doi: <https://doi.org/10.1016/j.arcontrol.2013.03.004>.
- [15] S. Peng and W. Shi, "Adaptive Fuzzy Output Feedback Control of a Nonholonomic Wheeled Mobile Robot," *IEEE Access*, vol. 6, pp. 43414–43424, 2018, doi: 10.1109/ACCESS.2018.2862163.
- [16] M. Beghini, D. W. Bertol, and N. A. Martins, "A robust adaptive fuzzy variable structure tracking control for the wheeled mobile robot: Simulation and experimental results," *Control Eng. Pract.*, vol. 64, pp. 27–43, 2017, doi: <https://doi.org/10.1016/j.conengprac.2017.04.006>.
- [17] P. Petrov and V. Georgieva, "Adaptive Velocity Control for a Differential Drive Mobile Robot," in *2018 20th International Symposium on Electrical Apparatus and Technologies (SIELA)*, 2018, pp. 1–4, doi: 10.1109/SIELA.2018.8447091.
- [18] M. Oliveira, A. Castro, T. Madeira, E. Pedrosa, P. Dias, and V. Santos, "A ROS framework for the extrinsic calibration of intelligent vehicles: A multi-sensor, multi-modal approach," *Rob. Auton. Syst.*, vol. 131, p. 103558, 2020, doi: <https://doi.org/10.1016/j.robot.2020.103558>.
- [19] R. Wang, Y. Li, J. Fan, T. Wang, and X. Chen, "A Novel Pure Pursuit Algorithm for Autonomous Vehicles Based on Salp Swarm Algorithm and Velocity Controller," *IEEE Access*, vol. 8, pp. 166525–166540, 2020, doi: 10.1109/ACCESS.2020.3023071.
- [20] J. Morales, J. L. Martínez, M. A. Martínez, and A. Mandow, "Pure-Pursuit Reactive Path Tracking for Nonholonomic Mobile Robots with a 2D Laser Scanner," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 1, p. 935237, 2009, doi: 10.1155/2009/935237.
- [21] Z. Wang, Y. Bai, J. Wang, and X. Wang, "Vehicle Path-Tracking Linear-Time-Varying Model Predictive Control Controller Parameter Selection Considering Central Process Unit Computational Load," *J. Dyn. Syst. Meas. Control*, vol. 141, no. 5, 2019, doi: 10.1115/1.4042196.
- [22] Q. Yao and Y. Tian, "A Model Predictive Controller with Longitudinal Speed Compensation for Autonomous Vehicle Path Tracking," *Appl. Sci.*, vol. 9, no. 22, 2019, doi: 10.3390/app9224739.
- [23] R. Marino, S. Scalzi, and M. Netto, "Nested PID steering control for lane keeping in autonomous vehicles," *Control Eng. Pract.*, vol. 19, no. 12, pp. 1459–1467, 2011, doi: <https://doi.org/10.1016/j.conengprac.2011.08.005>.
- [24] A. Bemporad, M. Morari, V. Dua, and E. N. Pistikopoulos, "The explicit linear quadratic regulator for constrained systems," *Automatica*, vol. 38, no. 1, pp. 3–20, 2002, doi: [https://doi.org/10.1016/S0005-1098\(01\)00174-1](https://doi.org/10.1016/S0005-1098(01)00174-1).
- [25] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A Predictive Controller for Autonomous Vehicle Path Tracking," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 1, pp. 92–102, 2009, doi: 10.1109/TITS.2008.2011697.

BIOGRAPHIES

OMUR AYDOGMUS received the B.S. (2000), M.S. (2004), and Ph.D. (2011) degree in electric-electronics engineering from the University of Firat. Since 2019, He has been a Professor in Department of Mechatronics Engineering, Faculty of Technology, Firat University, Elazig, Turkey. His research interests include electric machines and robotics.



GULLU BOZTAS received the B.S. (2011), M.S. (2015), and Ph.D. (2019) degree in electric-electronics engineering from the University of Firat. Since 2013, she has been a Research Assistant in Department of Electrical & Electronics Engineering, Faculty of Technology, Firat University, Elazig, Turkey. Her interests include power electronics and electric machines.

