

A Lyapunov based posture controller for a differential drive mobile robot

Boualem Kazed¹, Abderrezak Guessoum²

¹Electrical and Remote-Control Systems Laboratory (LAB SET), Department of Automatics and Electrical Engineering, Faculty of Technology, University of Blida, Blida, Algeria

²Signal Processing and Image Laboratory (LATS), Department of Electronics, Faculty of Technology, University of Blida, Blida, Algeria

Article Info

Article history:

Received Nov 13, 2023

Revised Dec 2, 2023

Accepted Dec 18, 2023

Keywords:

Hardware in the loop

Lyapunov controller

Posture control

Real-time control

Trajectory tracking

Unicycle robots

ABSTRACT

Driving a vehicle to a desired position and orientation is one of the most important problems that should be solved in most navigation systems. This paper describes a new complete design and hardware implementation of a two-level controller that will enable a differential drive mobile robot to reach any desired posture starting from any initial position. The first or low-level controller consists of a set of two proportional–integral–derivative (PID) controllers, running on an embedded system on board of the robot. These controllers provide the required voltages to the motors to make the left and right wheels of the robot rotate with the angular speeds computed by the second or high-level controller, running on a stationary PC system. This second controller is based on the Lyapunov stability theorem to derive two control laws for the kinematic model, used to transform the linear and angular speeds of the unicycle model in terms of left and right rotational speeds, required by the motors. As will be shown later, this architecture provides a very flexible way not only to tune the main controller parameters but also to get access and record all the system states.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Boualem Kazed

Department of Automatics and Electrical Engineering, Faculty of Technology, University of Blida

Route de Soumaa, Blida, Algeria

Email: kazed_boualem@univ-blida.dz

1. INTRODUCTION

Regardless of their mechanical structures, the main common problem for any mobile robot is motion control. Among all the existing configurations, differential drive mobile robots (DDMR) are widely used in indoor applications. This paper aims to present and implement a nonlinear, multivariable controller that will generate appropriate linear and angular speeds required by a mobile robot to move from a known initial position to a new target or desired posture. Solving this problem for a couple of consecutive points can therefore be considered as a very interesting alternative for the problem of trajectory tracking. Many of the recent research topics have tackled this problem using different control strategies, among these [1]–[3] have used the Lyapunov theory to derive the control laws allowing the robot to follow a predefined trajectory. Several other researchers [4]–[10] have proposed different approaches to solve the same problem. The reasons behind this interest are the possible real-world applications in various fields such as agriculture, shipping, cleaning, security, and many others. The main controller adopted in this study is based on the same state space model used in [11]–[16], from which we have defined an appropriate Lyapunov function and most importantly a new steering control law leading to a more straightforward proof for the asymptotic system stability, which was not clearly defined in the literature. In [17]–[19], two switching control laws are

proposed for the linear and angular velocities based on the relative initial and final postures. In [20], a Lyapunov-based controller derived from a function using the cartesian coordinates error is proposed. Widyotriatmo *et al.* [21] have suggested two separate Lyapunov function control schemes depending on the robot's closeness to the target position. Two different approaches namely time-varying oscillatory stabilizer and vector field orientation have been proposed in [22]. In [23], the feedback linearization technique has been utilized to derive a proportional-derivative (PD) controller for the position and orientation tracking problem. A linear controller adding an intermediate point to control the robot orientation has been used in [24]. A linearized kinematic model for the unicycle robot has been described and used to obtain a set of three proportional-integral-derivative (PID) controllers to solve the problem of path planning [25]. A finite time control technique has been adopted to stabilize the position of a mobile robot on a target posture using a bilinear structure for which two linear switching controllers are selected for each of the two subsystems [26]. The problem of posture stabilization using a passivity-based robust switching control has been described in [27]. In [28], a controller based on the transverse functions concept is presented, allowing the robot to track an omnidirectional moving target. Artificial intelligent-based approaches using reinforcement learning methods have been proposed to simulate the controlled robot movements to make it reach a predefined target position [29], [30]. An important feature of the present work is the modularity of the hardware implementation used to control the robotic platform specifically designed for this purpose. Unlike many of the results found in the literature, which are mainly based on simulation, the results shown in the related section are solely obtained from the experimental setup described in section 4.

The structure of this paper is organized as follows. Section 2 provides the detailed derivation of the DDMR kinematic model and its representation in terms of a state space model using polar coordinates. This model will then be used to design the high-level Lyapunov based controller, whose inputs are the desired robot posture data, and the outputs are the linear and angular velocities of the equivalent unicycle model. The experimental hardware platform used in this project is presented in Section 3, this also includes an overview of the low-level PID controllers used to generate the rotational reference speeds for each of the left and right direct current (DC) motors coupled with the wheels of the DDMR. All of the above topics will be put together in Section 4, this will include some real-life tests to evaluate the performance of the proposed approach. An overall conclusion will be drawn in Section 5.

2. SYSTEM MODELING AND CONTROL

2.1. Kinematic model of the DDMR

Figure 1 shows an example of a robot moving between two consecutive postures $P_1=(x_1, y_1, \theta_1)$ and $P_2=(x_2, y_2, \theta_2)$. Considering that these changes took place in a very short time Δt we can assimilate the portion of the robot trajectory as an arc of a circle centered on the instantaneous center of rotation (ICR) with radius R_C . The distance traveled by the left and right wheels are $S_L = R_L\alpha$ and $S_R = R_R\alpha$ respectively, at the same time the robot (center) covered the distance $S_C = R_C\alpha$. If we divide the terms of these equations by Δt we obtain the linear velocities; $v_L = R_L\dot{\alpha}$, $v_R = R_R\dot{\alpha}$, and $v = R_C\dot{\alpha}$. Note that $R_R = R_L + L$, L being the distance between the robot wheels, we will then have; $v_R - v_L = ((R_L + L) - R_L)\dot{\alpha} = L\dot{\alpha}$. Notice that $\alpha = \theta_2 - \theta_1 = \Delta\theta$, this will imply that $\dot{\alpha} = \dot{\theta}$. Considering ω_L and ω_R as the left and right wheels angular speeds we have $v_L = r\omega_L$ and $v_R = r\omega_R$, with $r=D/2$ being the radius of the robot wheels. Finally, we have $r\omega_R - r\omega_L = r(\omega_R - \omega_L) = L\dot{\theta}$, $\dot{\theta}$ is the angular velocity of the robot around the ICR, it will be denoted by Ω , this gives us (1).

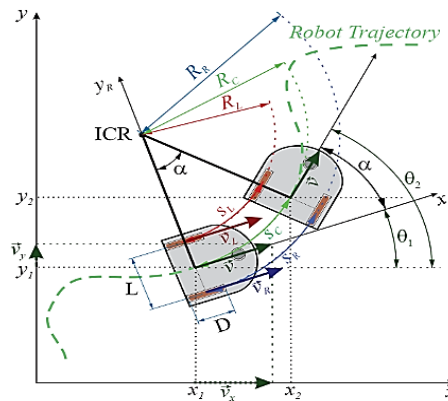


Figure 1. DDMR elementary displacements in a fixed frame (x, y)

$$\Omega = \frac{r}{L}(\omega_R - \omega_L) \quad (1)$$

From Figure 1, we have $R_C = R_L + L/2$ and $R_C = R_R - L/2$, therefore $2R_C = R_L + R_R$ or $R_C = (R_L + R_R)/2$ multiplying these terms by $\dot{\alpha}$ will give $R_C \dot{\alpha} = (R_L \dot{\alpha} + R_R \dot{\alpha})/2$ which can be rewritten as: $v = (v_L + v_R)/2$ or $v = (r\omega_L + r\omega_R)/2$, finally:

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

Equations (1) and (2) define the relationship between the DDMR driven by ω_L and ω_R and the unicycle model whose inputs are the linear and angular velocities v and the Ω respectively. The projections of the linear velocity v along the x and y axis are $v_x = \dot{x} = v \cos \theta$ and $v_y = \dot{y} = v \sin \theta$. Gathering these equations gives the kinematic model of the DDMR.

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = \Omega \end{cases} \quad (3)$$

2.2. State space model

Most of the known control approaches require a form of mathematical model for a system to be controlled, among these, state space representation is very often the preferred option. Figure 2 illustrates the situation where the robot is required to move between two postures P_1 and $P_2(x, y, \theta_2)$. To achieve this goal the robot needs to compute both the linear and angular speeds at each sample time. The translational part of the movement is obtained from the projection of the robot linear speed v , on the direction of the line joining the starting and final positions, this is designated by v_ρ . The orientation changes have to be made taking into account the fact that the total steering angle is a combination of two different angles; α is the part that makes the robot orient itself towards the final position and β the angle that directs it to the desired orientation θ_2 . The overall robot behavior can be modeled by (4).

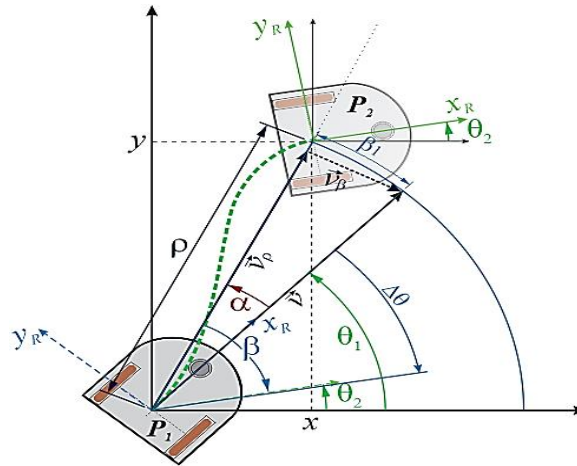


Figure 2. DDMR posture transition

$$\begin{cases} \rho = \sqrt{x^2 + y^2} \\ \alpha = \tan^{-1} \frac{y}{x} - \theta_1 \\ \beta = -\alpha + \Delta\theta \end{cases} \quad (4)$$

When the robot is moving towards its final goal the rate of change of the distance ρ is decreasing. This can be written as; $\dot{\rho} = v_\rho = -v \cos \alpha$. Figure 2 shows that β_1 is the portion of β that makes the robot align with the final orientation, therefore its rate of change $\dot{\beta}_1 = \dot{\beta}$ which implies that $\rho \dot{\beta} = v_\beta = -v \sin \alpha$. Using the third part of (4) we get $\alpha = -\beta + (\theta_2 - \theta_1)$, because θ_2 does not change and $\dot{\theta}_1 = \dot{\theta}$ we have $\dot{\alpha} = -\dot{\beta} - \dot{\theta} = -\dot{\beta} - \Omega$. Finally, the system can be represented by the state space model (5).

$$\begin{cases} \dot{\rho} = -v \cos \alpha \\ \dot{\beta} = -\frac{v}{\rho} \sin \alpha \\ \dot{\alpha} = \frac{v}{\rho} \sin \alpha - \Omega \end{cases} \quad (5)$$

2.3. Control design

From Figure 2, we can easily observe that the present control problem consists in finding a proper control law $u = f(v, \Omega)$ that will drive the state variables (ρ, β, α) to $(0, 0, 0)$. Let us use the quadratic form as a Lyapunov function (6).

$$V = \frac{1}{2}(\rho^2 + \alpha^2 + \beta^2) \quad (6)$$

Therefore

$$\begin{aligned} \dot{V} &= \rho\dot{\rho} + \alpha\dot{\alpha} + \beta\dot{\beta} = \rho(-v \cos \alpha) + \alpha\left(\frac{v}{\rho} \sin \alpha - \Omega\right) + \beta\left(-\frac{v}{\rho} \sin \alpha\right) \text{ or} \\ \dot{V} &= -v\rho \cos \alpha - \alpha\Omega + \alpha\frac{v}{\rho} \sin \alpha - \beta\frac{v}{\rho} \sin \alpha \end{aligned} \quad (7)$$

As already mentioned, v and Ω are the system control inputs. Let us use the law in (8) for the linear speed v , with k_ρ a positive constant.

$$v = k_\rho \rho \cos \alpha \quad (8)$$

This will give

$$\begin{aligned} \dot{V} &= -k_\rho \rho^2 \cos \alpha^2 - \alpha\Omega + \alpha\frac{k_\rho \rho \cos \alpha}{\rho} \sin \alpha - \beta\frac{k_\rho \rho \cos \alpha}{\rho} \sin \alpha \\ \dot{V} &= -k_\rho \rho^2 \cos \alpha^2 - \alpha\Omega + k_\rho \alpha \cos \alpha \sin \alpha - k_\rho \beta \cos \alpha \sin \alpha \end{aligned} \quad (9)$$

If we choose another positive constant k_α and the angular Ω speed as (10).

$$\Omega = k_\alpha \alpha + k_\rho (\alpha - \beta) \cos \alpha \frac{\sin \alpha}{\alpha} \quad (10)$$

We will have

$$\dot{V} = -k_\rho \rho^2 \cos \alpha^2 - k_\alpha \alpha^2 \quad (11)$$

Using the above control signals, the state space model (5) becomes (12).

$$\begin{cases} \dot{\rho} = -k_\rho \rho \cos \alpha^2 \\ \dot{\beta} = -k_\rho \cos \alpha \sin \alpha \\ \dot{\alpha} = -k_\rho (\alpha - \beta) \cos \alpha \frac{\sin \alpha}{\alpha} + k_\alpha \alpha + k_\rho \cos \alpha \sin \alpha \end{cases} \quad (12)$$

Using the Lyapunov theory if we can prove that \dot{V} is negative definite then the system described in (12) is asymptotically stable around the state $(\rho, \alpha, \beta) = (0, 0, 0)$. From (11) we have $\dot{V}(0, 0, \beta) = 0$, which would mean that \dot{V} is negative semi-definite, we can also observe that when $\alpha = 0$, $\frac{\sin \alpha}{\alpha} = 1$ and the third part of (12) becomes $\dot{\alpha} = k_\rho \beta$, but $\alpha = 0$ means that the system has reached its final state which infers that α remains constant and then $\dot{\alpha} = 0$ consequently $\beta = 0$. Finally, we can conclude that $\dot{V} = 0$ only if $(\rho, \alpha, \beta) = (0, 0, 0)$ which proves that the control laws (8) and (10) drive the system to an asymptotically stable state.

3. MOTOR SPEED CONTROL

To enable the robot, execute the required movements, the control laws previously defined must be transformed in terms of ω_L and ω_R , the robot left and right wheels angular velocities. These can be derived from (1) and (2) to get the following inverse unicycle model.

$$\omega_L = \frac{v}{r} - \frac{L\Omega}{2r} \quad (13)$$

$$\omega_R = \frac{v}{r} + \frac{L\Omega}{2r} \quad (14)$$

In order to ensure that ω_L and ω_R have actually the values defined in (13) and (14), these must be properly controlled using two separate closed loop subsystems as illustrated in Figure 3. Note that ω_{LM} and ω_{RM} are the measured speeds obtained from the derivatives of the angular positions, generated by the embedded quadrature encoders modules of the Microchip DSPic33fj64mc802 microcontroller.

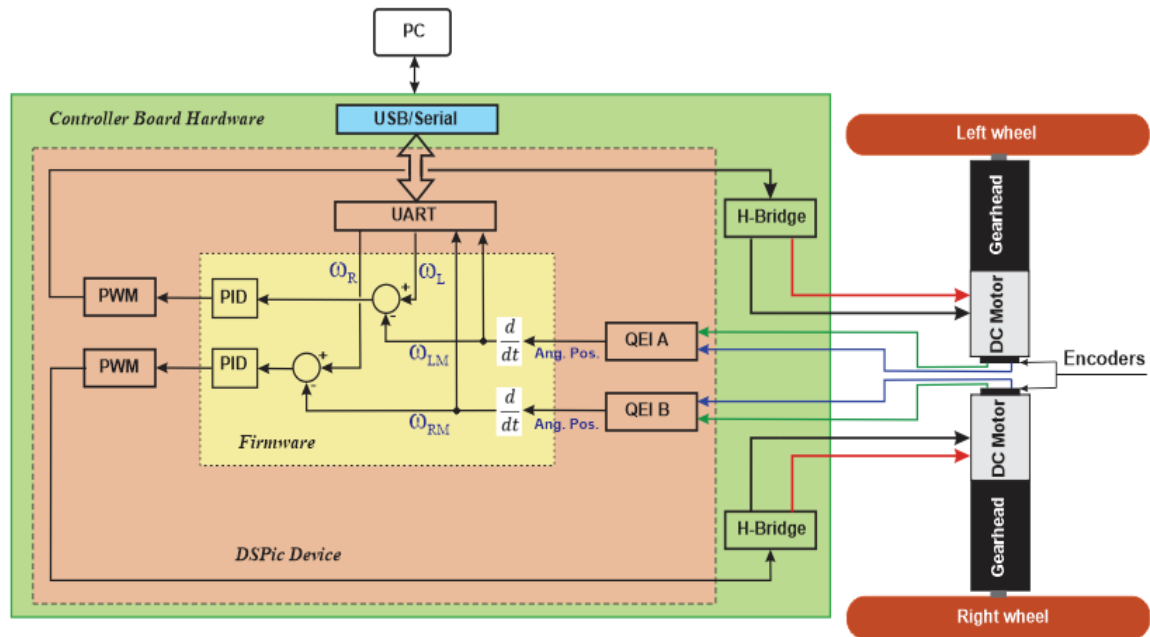


Figure 3. Embedded motor speed controller's block diagram

From the block diagram of Figure 3, we can distinguish 3 different colored zones. The inner yellow box shows that the two PID speed controllers have been implemented in a firmware flashed in the microcontroller program memory, the light-red rectangle represents the microcontroller with some of its embedded modules and the green area is the self-made electronic board, on which the microcontroller and all the required elements have been mounted. The aforementioned program receives the reference values of ω_L and ω_R through the serial universal asynchronous receiver-transmitter (UART) unit and delivers the PID control signals to the PWM modules, whose outputs are connected to the inputs of the L298N double H-bridge. Prior to implementing these controllers, their k_p , k_I and k_D parameters have been obtained using the following procedure: A mathematical model of one of the geared DC motors coupled with its wheels has been derived using the MATLAB system identification toolbox. Using the MATLAB/Simulink *pidentuner* toolbox, this model was then used to tune the PID parameters in a separate simulation program.

4. SIMULATION AND EXPERIMENTAL RESULTS

4.1. High-level controller implementation

Before showing the results obtained during the experiments we carried out, we have summarized the whole process of the proposed control scheme in a graphical format as shown in Figure 4. This bloc diagram clarifies the sequential operations involved after the introduction of the desired posture by the program user. As already mentioned, this program runs on top of the MATLAB/Simulink environment and is linked to the mobile robot through a serial communication hardware protocol, this is known as hardware in the loop (HIL) system. It should be noted that this is a real-time process and to make it work requires suitable sampling frequency and proper data formatting and parsing on both sides i.e., on the embedded control board and the computer side.

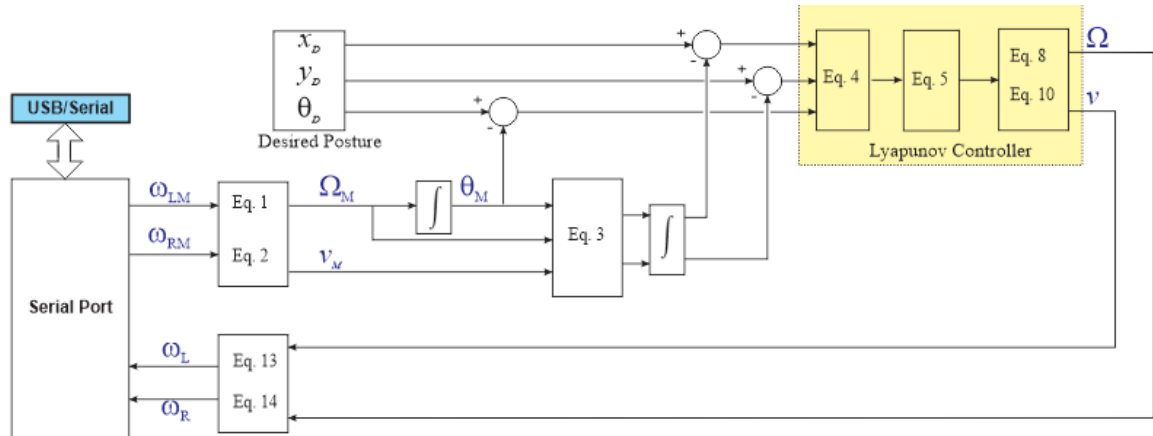


Figure 4. Bloc diagram of the high-level controller (M indices for measured signals)

4.2. Robotic platform

The constructed robotic platform used in this work is shown in Figure 5, most of the mechanical structure is made of plastic parts joined together with M8 and M6 nuts and bolts. From a purely theoretical point of view, such a physical system would require a dynamic model-based controller. This was not exactly the case in this study, where the kinematic model was the starting point of the designed controller. Nevertheless, the fact that the two PID speed controllers' parameters were tuned based on the dynamic model of the Maxon DC motor ref: 110147, coupled with an 84:1 reduction gearhead, including the robot wheels, the influence of the rest of the robot's physical parameters will not significantly change the robot overall behavior. Additionally, knowing that this robot will not be subject to very high accelerations, the kinematic model described above can be considered as good enough for developing a working controller.

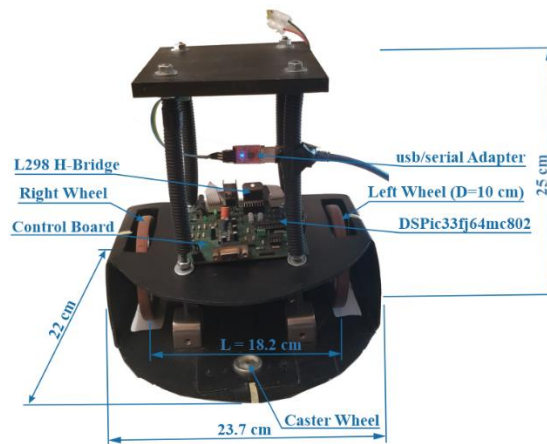


Figure 5. Picture of the self-made mobile robot

4.3. Results and discussions

The numerical values of the constants used in the experiments are listed in Table 1. The k_p and k_α parameters were obtained after a few iterations of trial-and-error testing using the proposed controller, applied to the complete system represented in Figures 3 and 4. For practical reasons, these parameters were tuned with the robot wheels rotating freely, without any contact with the ground.

Table 1. PID and Lyapunov controllers' parameters

k_p	k_l	k_D	k_p	k_α
0.0127	0.1573	0	0.8	1.8

Tuning these parameters with the robot moving on the ground would have been very difficult and very time-consuming. The selection of these final values was based on the observations of the system response in terms of both the error amplitudes and the time it takes to reach the desired posture. Several test runs have been executed as shown in Figure 6. As can be noticed these pictures were obtained by superimposing two photos; the first one being a sequence of several snapshots selected from recorded videos, taken during the robot movements. The second ones represent the robot trajectories and their successive positions, simulated using a MATLAB script that uses the robot postures actual data (x, y, θ) , recorded during the program executions. Figure 7 shows a sample of these plots, for the case f in Figure 6. From the top half of this figure, we can observe that the system states (ρ, α, β) (dashed lines) all converge to $(0, 0, 0)$ as the robot tracks the desired posture. This confirms the asymptotic stability of the state space system (12). From the bottom part of this figure, we can see that the measured left and right motors rotational speeds ω_{LM} and ω_{RM} are very close to ω_L and ω_R requested by the high-level Lyapunov controller. This proves that the two embedded PID controllers are performing very well. From the sample runs in Figure 6 we can also notice that when the robot is close to its final posture the linear speed v is always positive, this comes from its definition in (8), where the sign of $\cos \alpha$ is positive because, when this happens, α is close to zero. This may suggest that, for situations where the final goal is behind the starting position, as in the $f, g,$ and h cases in Figure 6, a better alternative would be to modify the control law (8) to make the robot move following a shorter trajectory.

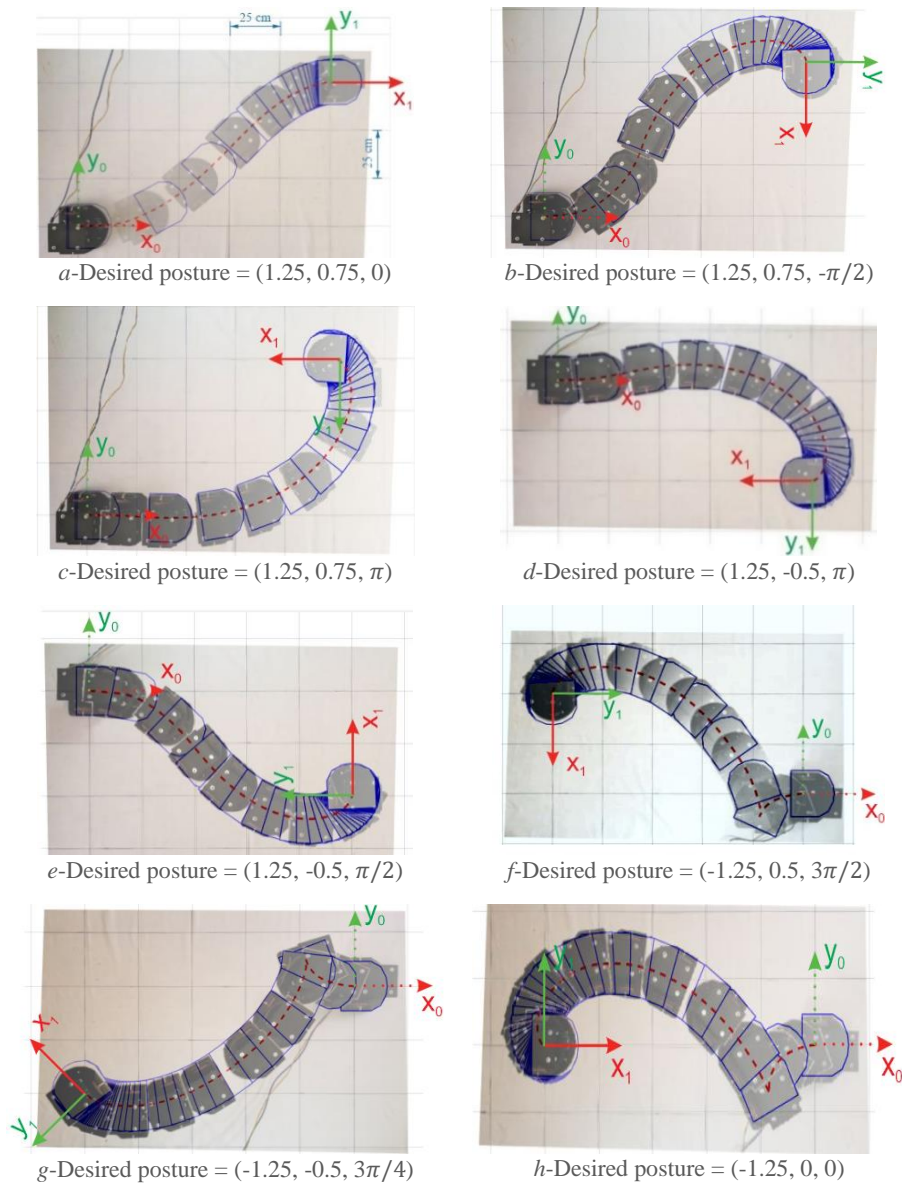


Figure 6. Sample tests of the robot movements to reach desired postures in the 4 quadrants

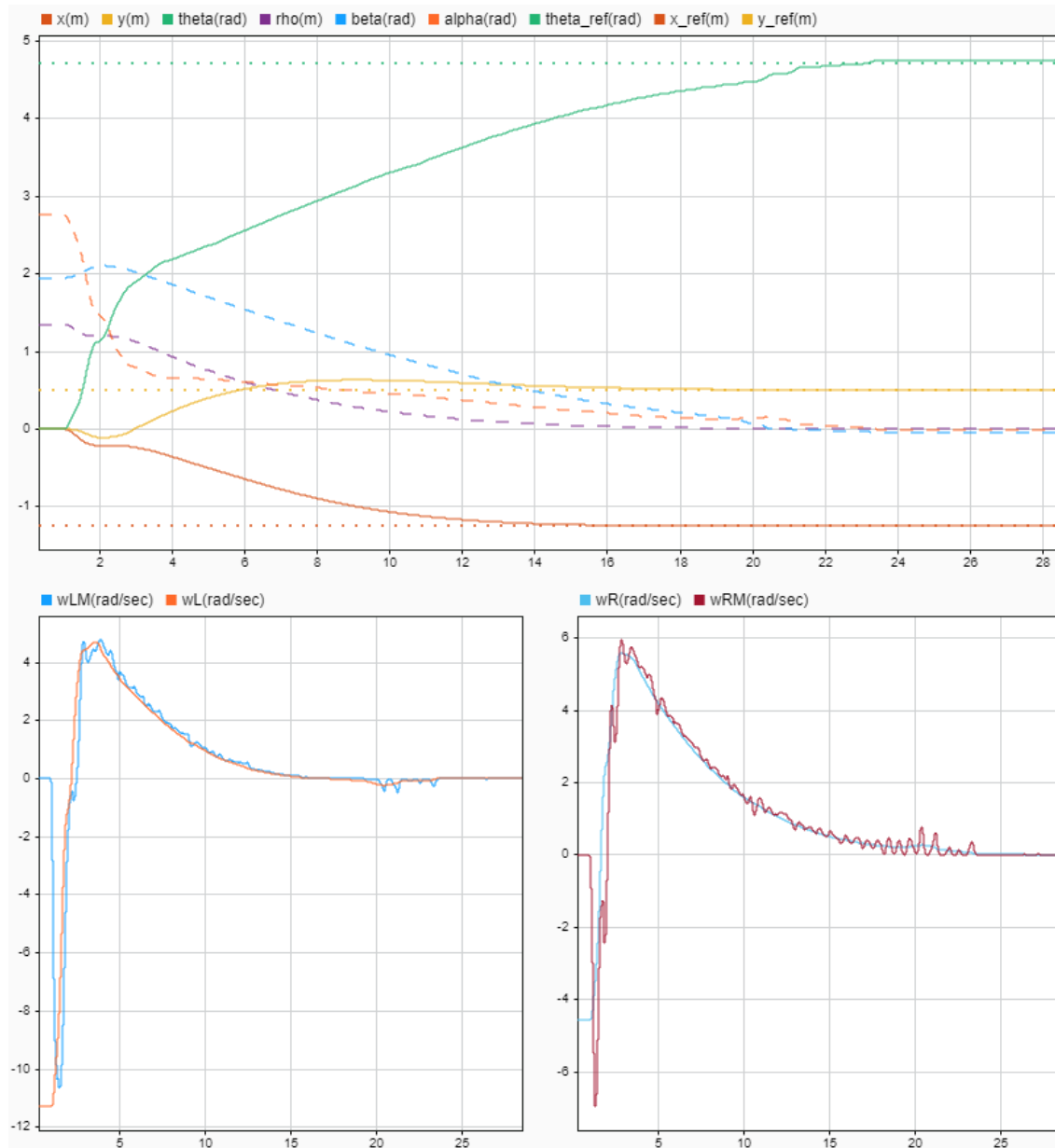


Figure 7. Time plots for the case where the desired posture is $(x, y, \theta) = (-1.25, 0.5, 3\pi/2)$

5. CONCLUSION

In this paper, a complete non-linear Lyapunov based controller was presented and applied to drive a unicycle-like robot to a desired posture. This was achieved on an experimental platform including the self-built mobile robot and a stationary personal computer (PC) system, on which this controller is implemented. The main reason for this architecture was not only to provide more flexibility for the controller's gains tuning, but also the possibility to implement and test any other suitable controller on the same PC. From the sample tests shown in Figures 6 and 7 and many others, the proposed controller has proven to give very good results in terms of the precision attained for both the final position and orientation. Further investigation to find a way to optimize the k_ρ and k_α gains of this controller, would probably result in a faster output response, especially for the final orientation, which takes much more time to stabilize than the set point position.




ACKNOWLEDGEMENTS

The authors would like to thank the support of the Algerian Ministry of Higher Education and Scientific Research through the PRFU project no. A01L08UN090120220001.




REFERENCES

- [1] M. Elsayed, A. Hammad, A. Hafez, and H. Mansour, "Real time trajectory tracking controller based on Lyapunov function for mobile robot," *International Journal of Computer Applications*, vol. 168, no. 11, pp. 1–6, Jun. 2017, doi: 10.5120/ijca2017914540.
- [2] A. M. Ali, A. Hussein, and M. Hassan, "Path tracking for nonholonomic mobile robot by using advance Lyapunov-based control laws," *International Journal of Engineering and Technology*, vol. 9, no. 2, pp. 200–206, Mar. 2019, doi: 10.14741/ijcet/v.9.2.1.
- [3] R. Kubo, Y. Fujii, and H. Nakamura, "Control Lyapunov function design for trajectory tracking problems of wheeled mobile robot," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 6177–6182, 2020, doi: 10.1016/j.ifacol.2020.12.1704.
- [4] L. Xiao-Feng, Z. Ren-Fang, and C. Guo-Ping, "A New non-time based tracking control for mobile robot," *International Journal of Robotic Engineering*, vol. 5, no. 1, Dec. 2020, doi: 10.35840/2631-5106/4125.
- [5] S. Cao, Y. Jin, T. Trautmann, and K. Liu, "Design and experiments of autonomous path tracking based on dead reckoning," *Applied Sciences (Switzerland)*, vol. 13, no. 1, p. 317, Dec. 2023, doi: 10.3390/app13010317.
- [6] J. Zhang, Q. Gong, Y. Zhang, and J. Wang, "Finite-time global trajectory tracking control for uncertain wheeled mobile robots," *IEEE Access*, vol. 8, pp. 187808–187813, 2020, doi: 10.1109/ACCESS.2020.3030633.
- [7] Y. M. Choi and J. H. Park, "Game-based lateral and longitudinal coupling control for autonomous vehicle trajectory tracking," *IEEE Access*, vol. 10, pp. 31723–31731, 2022, doi: 10.1109/ACCESS.2021.3135489.
- [8] N. Hassan and A. Saleem, "Neural network-based adaptive controller for trajectory tracking of wheeled mobile robots," *IEEE Access*, vol. 10, pp. 13582–13597, 2022, doi: 10.1109/ACCESS.2022.3146970.
- [9] M. J. Rabbani and A. Y. Memon, "Trajectory tracking and stabilization of nonholonomic wheeled mobile robot using recursive integral backstepping control," *Electronics (Switzerland)*, vol. 10, no. 16, p. 1992, Aug. 2021, doi: 10.3390/electronics10161992.
- [10] L. Song, J. Huang, Q. Liang, L. Nie, X. Liang, and J. Zhu, "Trajectory tracking strategy for sliding mode control with double closed-loop for lawn mowing robot based on ESO," *IEEE Access*, vol. 11, pp. 1867–1882, 2023, doi: 10.1109/ACCESS.2022.3166816.
- [11] M. Aicardi, G. Casalino, A. Bicchi, and A. Balestrino, "Closed loop steering of unicycle-like vehicles via Lyapunov techniques," *IEEE Robotics and Automation Magazine*, vol. 2, no. 1, pp. 27–35, Mar. 1995, doi: 10.1109/100.388294.
- [12] S. Kumar Malu and J. Majumdar, "Kinematics, localization and control of differential drive mobile robot," *Type: Double Blind Peer Reviewed International Research Journal Publisher: Global Journals Inc*, vol. 14, 2014.
- [13] T. H. Tran, M. D. Phung, T. T. Van Nguyen, and Q. V. Tran, "Stabilization control of the differential mobile robot using Lyapunov function and extended Kalman filter," *Vietnam Journals Online*, pp. 441–452, 2012.
- [14] C. C. de Wit, H. Khenouf, C. Samson, and O. J. Sordalen, "Nonlinear control design for mobile robots," in *World Scientific Series in Robotics and Intelligent Systems*, WORLD SCIENTIFIC, 1994, pp. 121–156. doi: 10.1142/9789814354301_0005.
- [15] K. Amar and S. Mohamed, "Stabilized feedback control of unicycle mobile robots," *International Journal of Advanced Robotic Systems*, vol. 10, no. 4, Art. no. 187, Jan. 2013, doi: 10.5772/51323.
- [16] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings., IEEE International Conference on Robotics and Automation*, 1990, pp. 384–389. doi: 10.1109/robot.1990.126006.
- [17] E. Fabregas *et al.*, "Simulation and experimental results of a new control strategy for point stabilization of nonholonomic mobile robots," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 8, pp. 6679–6687, Aug. 2020, doi: 10.1109/TIE.2019.2935976.
- [18] H. S. Shim and Y. G. Sung, "Stability and four-posture control for nonholonomic mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 148–154, Feb. 2004, doi: 10.1109/TRA.2003.819730.
- [19] H. S. Shim and Y. G. Sung, "A posture control for two wheeled mobile robots," *Transaction on Control Automation, and Systems Engineering*, vol. 2, no. 3, pp. 201–206, Sep. 2000, Accessed: Jan. 24, 2024. [Online]. Available: <https://www.ijcas.org/journal/view.html?spage=201&volume=2&number=3>
- [20] P. Panahandeh, K. Alipour, B. Tarvirdizadeh, and A. Hadi, "A kinematic Lyapunov-based controller to posture stabilization of wheeled mobile robots," *Mechanical Systems and Signal Processing*, vol. 134, Art. no. 106319, Dec. 2019, doi: 10.1016/j.ymsp.2019.106319.
- [21] A. Widjotriatmo, K. S. Hong, and L. H. Prayudhi, "Robust stabilization of a wheeled vehicle: Hybrid feedback control design and experimental validation," *Journal of Mechanical Science and Technology*, vol. 24, no. 2, pp. 513–520, Feb. 2010, doi: 10.1007/s12206-010-0105-1.
- [22] K. Kozłowski, J. Majchrzak, M. Michałek, and D. Pazderski, "Posture stabilization of a unicycle mobile robot - Two control approaches," in *Lecture Notes in Control and Information Sciences*, vol. 335, London: Springer London, 2006, pp. 25–54. doi: 10.1007/978-1-84628-405-2_2.
- [23] G. Oriolo, A. De Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: Design, implementation, and experimental validation," *IEEE Transactions on Control Systems Technology*, vol. 10, no. 6, pp. 835–852, Nov. 2002, doi: 10.1109/TCST.2002.804116.
- [24] and A. P. A. J. F. C. Vieira, A. A. D. Medeiros, P. J. Alsina, "Position and orientation control of a two-wheeled differentially driven nonholonomic mobile robot," in *Proceedings of the First International Conference on Informatics in Control, Automation and Robotics*, 2011, pp. 256–262. doi: 10.5220/0001138702560262.
- [25] P. K. Padhy, T. Sasaki, S. Nakamura, and H. Hashimoto, "Modeling and position control of mobile robot," in *International Workshop on Advanced Motion Control, AMC*, Mar. 2010, pp. 100–105. doi: 10.1109/AMC.2010.5464018.
- [26] M. Thomas, B. Bandyopadhyay, and L. Vachhani, "Posture stabilization of unicycle mobile robot using finite time control techniques," *IFAC-PapersOnLine*, vol. 49, no. 1, pp. 379–384, 2016, doi: 10.1016/j.ifacol.2016.03.083.
- [27] D. Y. Jung and S. Bang, "Posture stabilization of wheeled mobile robot based on passivity-based robust switching control with model uncertainty compensation," *Applied Sciences (Switzerland)*, vol. 9, no. 23, Art. no. 5233, Dec. 2019, doi: 10.3390/app9235233.
- [28] G. Artus, P. Morin, and C. Samson, "Tracking of an omnidirectional target with a unicycle-like robot: control design and experimental results," *Icar*, no. 4849, pp. 1468–1473, Jun. 2003.
- [29] F. Quiroga, G. Hermosilla, G. Farias, E. Fabregas, and G. Montenegro, "Position control of a mobile robot through deep reinforcement learning," *Applied Sciences (Switzerland)*, vol. 12, no. 14, Art. no. 7194, Jul. 2022, doi: 10.3390/app12147194.
- [30] G. Farias, G. Garcia, G. Montenegro, E. Fabregas, S. Dormido-Canto, and S. Dormido, "Position control of a mobile robot using reinforcement learning," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17393–17398, 2020, doi: 10.1016/j.ifacol.2020.12.2093.

BIOGRAPHIES OF AUTHORS

Boualem Kazed    received the Ing. d'état degree in electronics engineering from the National Polytechnics of Algiers in 1985 and an MPhil degree from the control engineering department of the University of Sheffield UK, in 1988. Since 1989 he has been teaching different graduate and postgraduate courses in electronics and control engineering, he has also supervised more than 160 final year projects related to real time data acquisition, control systems and robotics. He has been involved in many conferences and other activities related to educational robotics. His research interests include real-time control systems, mobile and industrial robotics. He can be contacted at email: kazed_boualem@univ-blida.dz.



Abderrezak Guessoum    received a BS degree in electronics from the Ecole Nationale Polytechnique d'Alger, Algeria, in 1976. He then went to Georgia Institute of Technology in Atlanta, Ga, USA, where he obtained first, a master's in electrical engineering in 1979, and then, a master's in applied mathematics in 1980 and finally a PhD in electrical engineering, in 1984. He worked as an Assistant Professor in the School of Computer Science at Jackson State University, Mississippi, USA, from 1984 to 1985. He has been with the Université de Blida, Algeria, as a professor in the department of Electronics, since 1988. Prof. Guessoum is the head of the Digital Signal and Image Processing Research Laboratory. His current research interests include digital filter design, adaptive filters, intelligent control, neural networks, genetic algorithms, robust control, metaheuristic optimization and fuzzy logic. He is the author of more than a hundred scientific articles. He can be contacted at: guessoum_abderrezak@univ-blida.dz.