IAES International Journal of Robotics and Automation (IJRA)
Vol. 15, No. 1, March 2026, pp. 33~42
ISSN: 2722-2586, DOI: 10.11591/ijra.v15i1.pp33-42

Experimental validation of a trajectory tracking controller for a two-wheeled mobile robot

Boualem Kazed[1], Abderrezak Guessoum[2]
[1]Electrical and Remote-Control Systems Laboratory (LAB SET), Department of Automatics and Electrical Engineering, Faculty of Technology, University of Blida, Blida, Algeria
[2]Signal Processing and Image Laboratory (LATSI), Department of Electronics, Faculty of Technology, University of Blida, Blida, Algeria

Article Info

Article history:

Received Aug 18, 2025
Revised Nov 15, 2025
Accepted Dec 15, 2025

Keywords:

Hardware in the loop
Lyapunov theory
Mobile robot
Real-time control
Trajectory tracking

ABSTRACT

One of the most important and challenging problems of any kind of autonomous mobile robot is the ability to accurately control its onboard actuators, enabling it to fulfill a specified task. In the case of a two-wheeled mobile robot, this can only be achieved through a pair of adequate steering control signals. The main goal of this paper is to design a nonlinear multivariable controller allowing a self-made mobile robot prototype to track a prescribed trajectory. The basic principle of this control approach uses the Lyapunov theory as a primary tool to derive two steering control laws, making a three-state error vector converge to zero. Tuning the proposed controller parameters is carried out using an equivalent dynamic simulated model. This controller is then applied to generate the resulting command signals to the actual robot. This is achieved through a real-time high-speed serial communication between a stationary personal computer (PC), on which a MATLAB/Simulink version of this controller is performing, and an onboard Microchip 16 bits dsPIC33FJ64MC802 microcontroller running a firmware that takes care of all the data exchange with the connected PC and a set of two proportional integral derivative (PID) controllers ensuring that the rotational speeds of the robot wheels are kept very close to those required by the main controller, running on this PC. The performance of the proposed controller is evaluated using two different shaped trajectories. These tests show that the robot is able to gradually follow the required path with minimal lateral error. The robustness of this controller is demonstrated through its capability to reject external disturbances triggered during these experimental tests.

Corresponding Author:

Boualem Kazed
Department of Automatics and Electrical Engineering, Faculty of Technology, University of Blida
1 Route de Soumaa, Blida, Algeria
Email: kazed_boualem@univ-blida.dz

1. INTRODUCTION

Trajectory tracking is one of the most studied topics related to mobile robotics. This is probably motivated by the large number of applications for industrial and social needs in real-world life. This paper is meant to add another contribution to this field by going through a detailed explanation of the overall system modeling and control. The basic principle of the present work which was first introduced in [1], has been extensively reused in the literature, as in [2]–[10] and many others. The main common feature of these papers is the fact that they were all based on the same three tuning parameters controller. Using this same mathematical model, the authors of [11]–[17] have proposed different control strategies to achieve the same

task. This was also the case in [18] where the authors have combined a neural network to adapt the kinematic controller parameters to those of a reference model subsystem, including a linear quadratic regulator (LQR). Another controlling approach has been adopted in [19]. In this particular case, the proposed controller was composed of two distinct proportional integral derivative (PID) controllers, respectively for the translational and rotational motion of the robot. Based on the same model the authors of [20] have used a sliding mode controller to track linear and circular shaped trajectories. A linear quadratic regulator (LQR) based controller has also been utilized in [21]. A learning mechanism to update the feedforward action of the main controller has been designed to gradually minimize the tracking error between the robot and the prescribed trajectory [22]. The authors of [23] have proposed a neural network-based controller to enhance the performance of a genetic algorithm (GA) optimized nonlinear PID controller parameters. Another approach using a fuzzy logic based sliding mode control technique has been proposed in [24]. The problem of trajectory tracking for a car-like Autonomous Vehicle has been the focus of the work presented in [25]. In this study, the control strategy was based on a "linear parameter varying" (LPV) system for which the parameters of a model predictive controller (MPC) were optimized through a particle swarm optimization (PSO) technique. Trajectory tracking in 3D space is also one of the most attracting topics in the literature, among the many proposed controlling approaches the authors of [26] have designed a neural fuzzy based PID controller to make an unmanned aerial vehicle (UAV) follow a predefined trajectory. Aiming to solve the same problem, the authors in [27] have adopted a "Super twisting sliding mode controller" combined with a fuzzy PID surface to adjust the gain parameters and thereby reducing the chattering effect induced by this kind of control technique. As previously mentioned, the main focus of the present work is to design a controller enabling the mobile robot to follow a prescribed trajectory, in an obstacle-free environment. Therefore, this is only a partial solution to the more global problem of motion planning, which includes path planning, to generate a succession of waypoints that the robot should go through, including the time samples at which the robot is supposed to reach these waypoints. Path planning is therefore another important problem that has been taken care of in many research works. This was the case in [28], where the authors have suggested a "Bidirectional rapidly-exploring random tree star", integrated with a "Dynamic window approach" and an "Adaptive monte Carlo localization" technique.

In this paper, using simple geometric homogeneous transformation properties, an error dynamic model will be derived and used to obtain a two-component nonlinear control scheme for both the linear and angular speeds of the mobile robot. As for any other controller, the end goal of its design is to implement a customized version of it on a real hardware platform. In the particular case of a mobile robot, this could prove to be very time-consuming, especially when trying to tune the unknown parameters. This is caused by the fact that we need to update the firmware of the onboard system every time a change of these parameters is required. One of the key features of the present work is to address this problem using a hardware-in-the-loop architecture, allowing to debug the controlling part on a stationary PC, while leaving unchanged the firmware running on the moving robot. What makes this deployment even faster is the ability to carry out the controller tuning phase using a robot dynamic model as part of a complete simulation program, before replacing this model with the connected hardware platform.

The content of this paper can be summarized as follows; In section 2, a comprehensive and detailed description of the kinematic model is given. The resulting, nonlinear state space model is then used to design a controller for steering the robot towards the desired path. The results of some illustrative examples, used to test the performance of the proposed controller, will be discussed in section 3. The most important features of the proposed control system will be summarized in section 4.

2. MODELING AND CONTROL

2.1. Path following strategy

One of the most popular methods used to make a robot track a predefined trajectory, consists of designing a controller capable of driving this robot towards another virtual robot that is perfectly following the target or desired trajectory. This approach is illustrated in Figure 1, where, in this case the virtual robot is designated by the letter B, whereas the controlled robot is labeled robot A. To achieve this goal, we first need to express the posture (position and orientation) of robot B relative to the frame attached to robot A. In the following, we will use the notation H_M^F to define a transformation of a moving (M) frame with respect to a fixed (F) frame. Considering that the postures of robot A and B can be represented by their respective transformations in the world or fixed frame (0) we have:

$$H_A^0 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & x_1 \\ \sin\theta_1 & \cos\theta_1 & y_1 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

$$H_B^0 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & x_2 \\ \sin\theta_2 & \cos\theta_2 & y_2 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

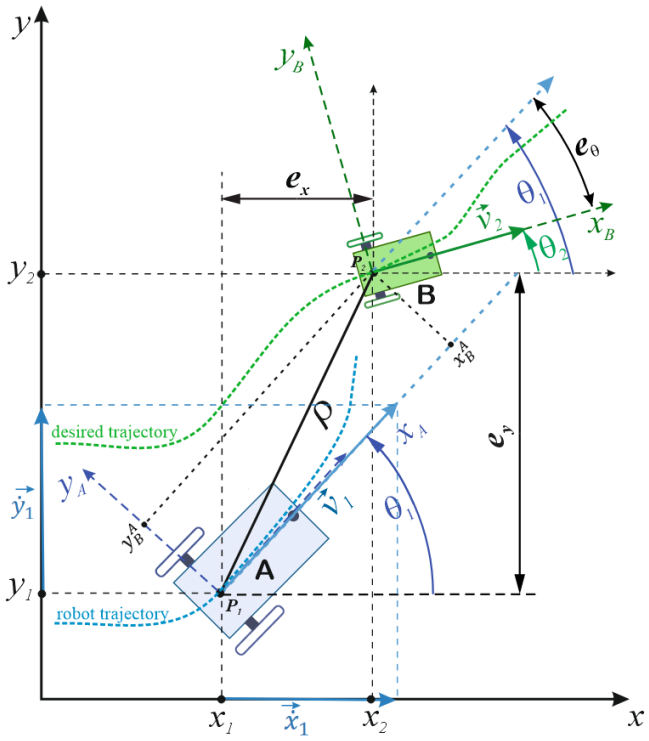

Figure 1. Sample postures of the actual (grey) and virtual robot (green)

The transformation matrix H_B^0 moves and rotates the frame (0) from (0,0,0) to the posture (x_2, y_2, θ_2). The fixed frame's transformation matrix can be written as $H_0^0 = I$, with I being the identity matrix. Rewriting H_B^0 as $H_B^0 = H_0^0 H_B^0$ expresses the fact that H_B^0 can be viewed as an operator applied on the frame (0) to move it and places it on frame (2), attached to robot B. If we change the starting posture and make it on frame (1) which is attached to robot A, H_0^0 will be replaced by H_A^0, therefore H_B^0 can also be expressed as $H_B^0 = H_A^0 H_B^A$, if we pre-multiply both sides by $(H_A^0)^{-1}$ we get:

$$H_B^A = (H_A^0)^{-1} H_B^0 = \begin{bmatrix} \cos\theta_1 & \sin\theta_1 & -x_1\cos\theta_1 - y_1\sin\theta_1 \\ -\sin\theta_1 & \cos\theta_1 & x_1\sin\theta_1 - y_1\cos\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & x_2 \\ \sin\theta_2 & \cos\theta_2 & y_2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$H_B^A = \begin{bmatrix} \cos(\theta_2 - \theta_1) & -\sin(\theta_2 - \theta_1) & (x_2 - x_1)\cos\theta_1 + (y_2 - y_1)\sin\theta_1 \\ \sin(\theta_2 - \theta_1) & \cos(\theta_2 - \theta_1) & (y_2 - y_1)\cos\theta_1 - (x_2 - x_1)\sin\theta_1 \\ 0 & 0 & 1 \end{bmatrix} \quad (3)$$

We can easily notice that the matrix obtained in (3) represents a homogeneous transformation with an angular rotation of $\theta_B^A = (\theta_2 - \theta_1)$ around the z axis, perpendicular and pointing out of the page, and two translations: $x_B^A = (x_2 - x_1)\cos\theta_1 + (y_2 - y_1)\sin\theta_1$ and $y_B^A = (y_2 - y_1)\cos\theta_1 - (x_2 - x_1)\sin\theta_1$, along the x_A and y_A axis respectively. Taking the derivatives of these linear and angular positions, we get:

$$\begin{aligned} \dot{x}_B^A &= (\dot{x}_2 - \dot{x}_1)\cos\theta_1 - (x_2 - x_1)\dot{\theta}_1\sin\theta_1 + (\dot{y}_2 - \dot{y}_1)\sin\theta_1 + (y_2 - y_1)\dot{\theta}\cos\theta_1 \\ \dot{y}_B^A &= -(\dot{x}_2 - \dot{x}_1)\sin\theta_1 - (x_2 - x_1)\dot{\theta}_1\cos\theta_1 + (\dot{y}_2 - \dot{y}_1)\cos\theta_1 - (y_2 - y_1)\dot{\theta}\sin\theta_1 \\ \dot{\theta}_B^A &= \dot{\theta}_2 - \dot{\theta}_1 \end{aligned} \quad (4)$$

In matrix form equation (4) can be written as (5),

$$\begin{bmatrix}\dot{x}_B^A\\ \dot{y}_B^A\\ \dot{\theta}_B^A\end{bmatrix}=\begin{bmatrix}-\dot{\theta}_1\sin\theta_1 & \dot{\theta}_1\cos\theta_1 & 0\\ -\dot{\theta}_1\cos\theta_1 & -\dot{\theta}_1\sin\theta_1 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}x_2-x_1\\ y_2-y_1\\ \theta_2-\theta_1\end{bmatrix}+\begin{bmatrix}\cos\theta_1 & \sin\theta_1 & 0\\ -\sin\theta_1 & \cos\theta_1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}_2-\dot{x}_1\\ \dot{y}_2-\dot{y}_1\\ \dot{\theta}_2-\dot{\theta}_1\end{bmatrix} \quad (5)$$

which can further be expressed as (6).

$$\begin{bmatrix}\dot{x}_B^A\\ \dot{y}_B^A\\ \dot{\theta}_B^A\end{bmatrix}=\begin{bmatrix}-\dot{\theta}_1\sin\theta_1 & \dot{\theta}_1\cos\theta_1 & 0\\ -\dot{\theta}_1\cos\theta_1 & -\dot{\theta}_1\sin\theta_1 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}e_x\\ e_y\\ e_\theta\end{bmatrix}+\begin{bmatrix}\cos\theta_1 & \sin\theta_1 & 0\\ -\sin\theta_1 & \cos\theta_1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}_2\\ \dot{y}_2\\ \dot{\theta}_2\end{bmatrix}-\begin{bmatrix}\cos\theta_1 & \sin\theta_1 & 0\\ -\sin\theta_1 & \cos\theta_1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}_1\\ \dot{y}_1\\ \dot{\theta}_1\end{bmatrix} \quad (6)$$

From Figure 1, $\dot{x}_1$ and $\dot{y}_1$ can be replaced as follows $\dot{x}_1=v_1\cos\theta_1$ and $\dot{y}_1=v_1\sin\theta_1$, the angular speed of the robot around the vertical axis *z*, $\dot{\theta}_1$ will be denoted as Ω_1.

Using these expressions the last term of (6) becomes (7).

$$\begin{bmatrix}\cos\theta_1 & \sin\theta_1 & 0\\ -\sin\theta_1 & \cos\theta_1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}\dot{x}_1\\ \dot{y}_1\\ \dot{\theta}_1\end{bmatrix}=\begin{bmatrix}\dot{x}_1\cos\theta_1+\dot{y}_1\sin\theta_1\\ -\dot{x}_1\sin\theta_1+\dot{y}_1\cos\theta_1\\ \dot{\theta}_1\end{bmatrix}=\begin{bmatrix}v_1\cos\theta_1\cos\theta_1+v_1\sin\theta_1\sin\theta_1\\ -v_1\cos\theta_1\sin\theta_1+v_1\sin\theta_1\cos\theta_1\\ \dot{\theta}_1\end{bmatrix}=\begin{bmatrix}v_1\\ 0\\ \Omega_1\end{bmatrix} \quad (7)$$

Similarly noting that $\dot{x}_2=v_2\cos\theta_2$ and $\dot{y}_2=v_2\sin\theta_2$ equation (6) can be reduced to:

$$\begin{bmatrix}\dot{x}_B^A\\ \dot{y}_B^A\\ \dot{\theta}_B^A\end{bmatrix}=-\dot{\theta}_1\begin{bmatrix}\sin\theta_1 & -\cos\theta_1 & 0\\ \cos\theta_1 & \sin\theta_1 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}e_x\\ e_y\\ e_\theta\end{bmatrix}-\begin{bmatrix}v_1\\ 0\\ \Omega_1\end{bmatrix}+\begin{bmatrix}\cos\theta_1 & \sin\theta_1 & 0\\ -\sin\theta_1 & \cos\theta_1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}v_2\cos\theta_2\\ v_2\sin\theta_2\\ \Omega_2\end{bmatrix} \quad (8)$$

Which can be rewritten as:

$$\begin{bmatrix}\dot{x}_B^A\\ \dot{y}_B^A\\ \dot{\theta}_B^A\end{bmatrix}=-\dot{\theta}_1\begin{bmatrix}\sin\theta_1 & -\cos\theta_1 & 0\\ \cos\theta_1 & \sin\theta_1 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}e_x\\ e_y\\ e_\theta\end{bmatrix}-\begin{bmatrix}v_1\\ 0\\ \Omega_1\end{bmatrix}+\begin{bmatrix}v_2\cos\theta_1\cos\theta_2+v_2\sin\theta_1\sin\theta_2\\ -v_2\sin\theta_1\cos\theta_2+v_2\cos\theta_1\sin\theta_2\\ \Omega_2\end{bmatrix}$$

And then:

$$\begin{bmatrix}\dot{x}_B^A\\ \dot{y}_B^A\\ \dot{\theta}_B^A\end{bmatrix}=-\dot{\theta}_1\begin{bmatrix}\sin\theta_1 & -\cos\theta_1 & 0\\ \cos\theta_1 & \sin\theta_1 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{bmatrix}e_x\\ e_y\\ e_\theta\end{bmatrix}-\begin{bmatrix}v_1\\ 0\\ \Omega_1\end{bmatrix}+\begin{bmatrix}v_2\cos(\theta_2-\theta_1)\\ v_2\sin(\theta_2-\theta_1)\\ \Omega_2\end{bmatrix}$$

From the third column of the transformation matrix (3), we have:

$$x_B^A=(x_2-x_1)\cos\theta_1+(y_2-y_1)\sin\theta_1=e_x\cos\theta_1+e_y\sin\theta_1$$
$$y_B^A=(y_2-y_1)\cos\theta_1-(x_2-x_1)\sin\theta_1=e_y\cos\theta_1-e_x\sin\theta_1$$

Finaly the error dynamics of robot A relative to robot B can expressed as:

$$\begin{bmatrix}\dot{x}_B^A\\ \dot{y}_B^A\\ \dot{\theta}_B^A\end{bmatrix}=-\Omega_1\begin{bmatrix}-y_B^A\\ x_B^A\\ 0\end{bmatrix}-\begin{bmatrix}v_1\\ 0\\ \Omega_1\end{bmatrix}+\begin{bmatrix}v_2\cos e_\theta\\ v_2\sin e_\theta\\ \Omega_2\end{bmatrix}=\begin{bmatrix}\Omega_1 y_B^A-v_1+v_2\cos e_\theta\\ -\Omega_1 x_B^A+v_2\sin e_\theta\\ \Omega_2-\Omega_1\end{bmatrix} \quad (9)$$

For more convenience let us define the state vector $X=\begin{bmatrix}X_1\\ X_2\\ X_3\end{bmatrix}=\begin{bmatrix}x_B^A\\ y_B^A\\ \theta_B^A\end{bmatrix}$. This gives us the following state space system in (10).

$$\begin{cases}\dot{X}_1=\Omega_1X_2-v_1+v_2\cos X_3\\ \dot{X}_2=-\Omega_1X_1+v_2\sin X_3\\ \dot{X}_3=\Omega_2-\Omega_1\end{cases} \quad (10)$$

2.2. Control design

To make robot A track the desired trajectory, the linear and angular speeds v_1 and Ω_1, need to be properly evaluated. To this end, let us define the positive definite function V (11) as a Lyapunov function candidate.

$$V = \frac{1}{2}(X_1^2 + X_2^2) + 1 - \cos X_3 \tag{11}$$

The term $(1 - \cos X_3)$ insures that V is always positive definite with respect to the orientation X_3; $V(X_1, X_2, X_3) > 0 \quad \forall (X_1, X_2, X_3) \neq (0,0,0) \quad V(0,0,0) = 0$, the derivative of V is obtained as (12).

$$\begin{aligned} \dot{V} &= X_1\dot{X}_1 + X_2\dot{X}_2 + \dot{X}_3 \sin X_3 \\ \dot{V} &= X_1(\Omega_1 X_2 - v_1 + v_2 \cos X_3) + X_2(-\Omega_1 X_1 + v_2 \sin X_3) + (\Omega_2 - \Omega_1) \sin X_3 \\ \dot{V} &= X_1(v_2 \cos X_3 - v_1) + (v_2 X_2 + \Omega_2 - \Omega_1) \sin X_3 \end{aligned} \tag{12}$$

If we choose $(v_2 \cos X_3 - v_1) = -K_1 X_1$, and $(v_2 X_2 + \Omega_2 - \Omega_1) = -K_2 \sin X_3$, with $K_1, K_2 > 0$ some tuning parameters, we get control laws in (13).

$$\begin{cases} v_1 = K_1 X_1 + v_2 \cos X_3 \\ \Omega_1 = X_2 v_2 + \Omega_2 + K_2 \sin X_3 \end{cases} \tag{13}$$

With these choices (12) reduces to (14).

$$\dot{V} = -K_1 X_1^2 - K_2 \sin^2 X_3 \tag{14}$$

From (14) we can observe that $\dot{V}(0, X_2, 0) = 0$ which means that $\dot{V}$ is negative semi-definite. In this particular case, considering the fact the second terms of the system described by the differential equations (10) are time-varying, we only need to prove that this system is marginally or locally stable. This comes from the fact that the most important role of the controller is to drive the actual robot towards the target trajectory, keeping it very close to the virtual robot. Therefore, if we could prove that the controllers (13) provide a locally stable solution for the system (10) this problem would then be solved. Using the Lyapunov theory and the Barbalat's lemma [29] a sufficient condition is to prove that the second derivative of the proposed Lyapunov function V is bounded. From (14) we have:

$$\ddot{V} = -2K_1 X_1 \dot{X}_1 - 2K_2 \dot{X}_3 \cos X_3$$

Replacing $\dot{X}_1$ and $\dot{X}_3$ from (10) and using $v_1$and Ω_1 defined in (13) we get (15).

$$\ddot{V} = -2K_1 X_1((X_2 v_2 + \Omega_2 + K_2 \sin X_3)X_2 - K_1 X_1) - 2K_2(X_2 v_2 - K_2 \sin X_3) \cos X_3 \tag{15}$$

Except from the constants K_1 and K_2 all elements of (15) depend on the state variables X_1, X_2 and X_3. As already mentioned, $\dot{V} \leq 0$ this means that the Lyapunov function V defined in (11) is non increasing or bounded, therefore X_1, X_2 and X_3 are bounded, which in turn proves that $\ddot{V}$ is also bounded.

3. RESULTS AND DISCUSSIONS

3.1. Implementation details

The overall architecture of the adopted control system is summarized in Figure 2. From this bloc diagram we can see that, in addition to the proposed main controller, we also have two PID controllers specifically designed to make the left and right robot wheels rotate with the angular speeds ω_L and ω_R, required by the inverse kinematics bloc. It should be noted that these PID controllers have been tuned taking into account the actual robot dynamics, including the wheels attached to the shaft of the driving DC gearmotors. The measured angular speeds ω_{LM} and ω_{RM}, obtained from the actual robot or its equivalent model are then fed to the forward kinematics bloc, which in turn delivers the measured linear and angular speeds v_{1M} and Ω_{1M} of the robot chassis. These two signals are then used to close the outer loop making the whole system work as described.

3.2. Experimental results

After a few trial-and-error settings in simulation, the proposed controller parameters $K_1$and K_2 have been set to the values listed in Table 1. The performance of this controller has then been tested through

several scenarios, among which the results of an example using a Lissajous curve type trajectory, defined by the parametric equations: $x_2(t) = 2(\cos\frac{\pi}{50}t - 1) + 0.25$ and $y_2(t) = 2\sin\frac{\pi}{25}t$, are shown in Figures 3 and 4. A visual inspection of theses plots shows that this tracking experiment has been successfully achieved. After less than 20 seconds, the actual robot trajectory was nearly superimposed on the reference path drawn by the virtual robot. This can be verified in the second plot of Figure 3, in which the lateral error is shown to be very close to zero. As can be observed in Figure 4, the actual robot trajectory has been slightly deviated from the reference path after the position indicated by the (-2.4,1.23) coordinates. This is the result of an external disturbance caused by a manual hard push on the right wheel, preventing it from rotating freely during a short period, starting at the 70 second time sample. From the two last plots of Figure 3, one can see that the consequences of slowing down the right wheel has a direct effect on the left wheel, meaning that the proposed outer loop controller has been able to positively react against this unexpected perturbation, allowing the robot to be back on the reference path a few seconds later. This is also due to the important role played by the two PID controllers, which are responsible for matching the measured rotational speeds of both wheels with those computed by the main controller. This can also be verified in the last two plots of Figure 3, which shows that these requirements are effectively fulfilled.

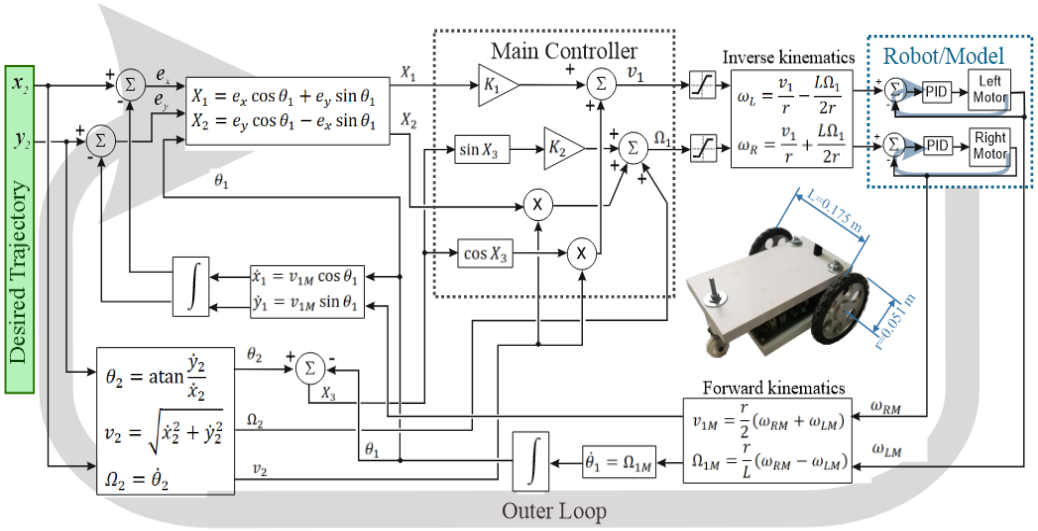

Figure 2. Block diagram of the complete control system

Table 1. PID and trajectory controllers' parameters

k_P	k_I	k_D	K_1	K_2
0.0127	0.1573	0	0.1	2.5

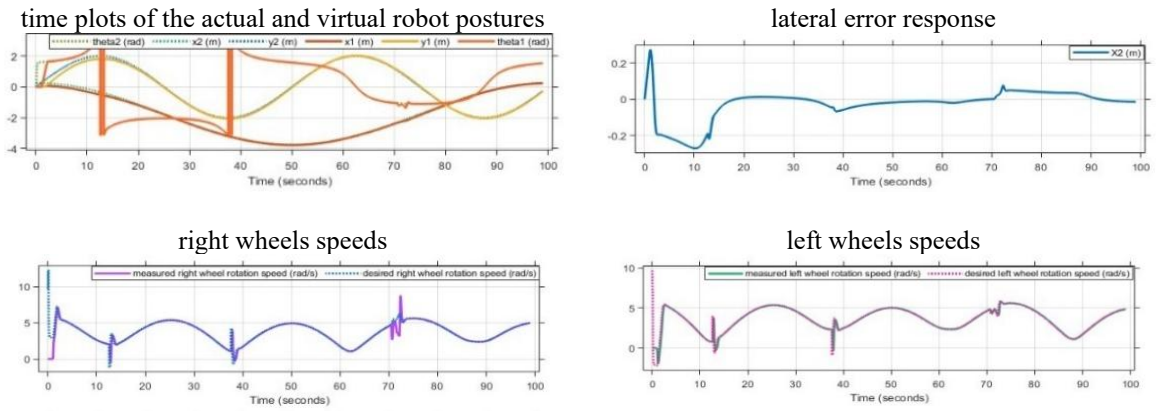

Figure 3. Real-time data acquired during the tracking of a Lissajous type trajectory

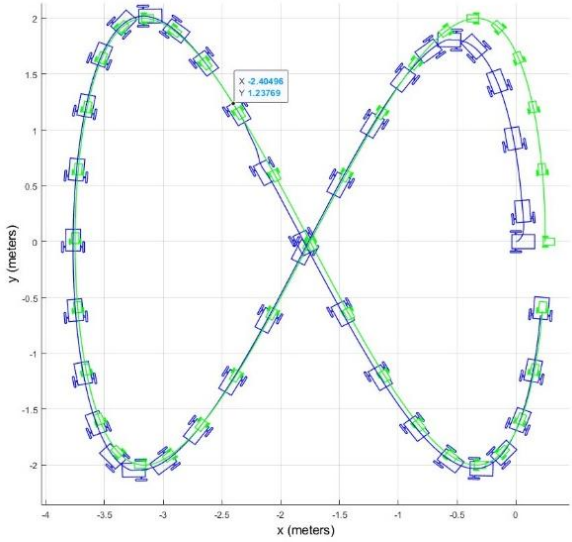

Figure 4. Experimental results for a Lissajous type trajectory tracking with external disturbances

In a second test, where robot has been moving on 0.25-meter grid marked surface, the snapshots shown in Figure 5 have been selected from a 150 seconds video recorded during this experiment. Figure 6 gives a more detailed view of some of the signals acquired during this process. For this example, the robot reference trajectory is a hypotrochoid defined by the parametric equations: $x_2(t) = 0.36\cos 0.2t + 0.164\cos 0.3t$ and $y_2(t) = 0.36\sin 0.2t - 0.164\sin 0.3t$. As can be noticed in Figure 5, this tracking experiment shows that after an initial transition of about 60 seconds, during which the robot has almost completed the first round, passing through the positions labeled 0 to 7, the lateral error X_2 has been gradually decreasing making the robot move very close to the target trajectory. From the upper part of Figure 6, we can see how the robot position and orientation get closer to those of the virtual robot during this sample test. The second part of Figure 6 shows the output of the proposed controller v_1 and ω_1, after having been limited through a pair of saturating blocs. These have been inserted to avoid letting the rotational speeds of the left and right geared motors shafts exceed their limit of $|\omega_{Max}| = 9.9\ rad/sec$, this is the case when the voltage applied to the motors reach the nominal value of $\pm 12\ volts$. From the forward kinematics equations one can easily get the corresponding maximum values of the linear and angular speeds of the robot chassis, these are given by: $v_{1Max} = 0.5\ m/sec$ and $\Omega_{1Max} = 5.5\ rad/sec$ respectively.

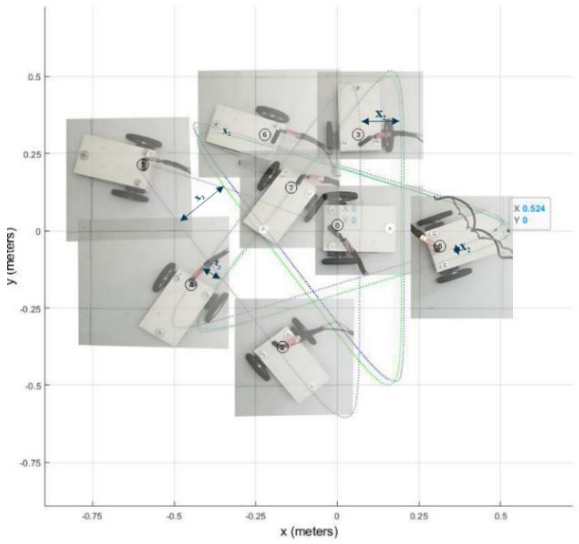

Figure 5. Hypotrochoid trajectory tracking: sample robot postures

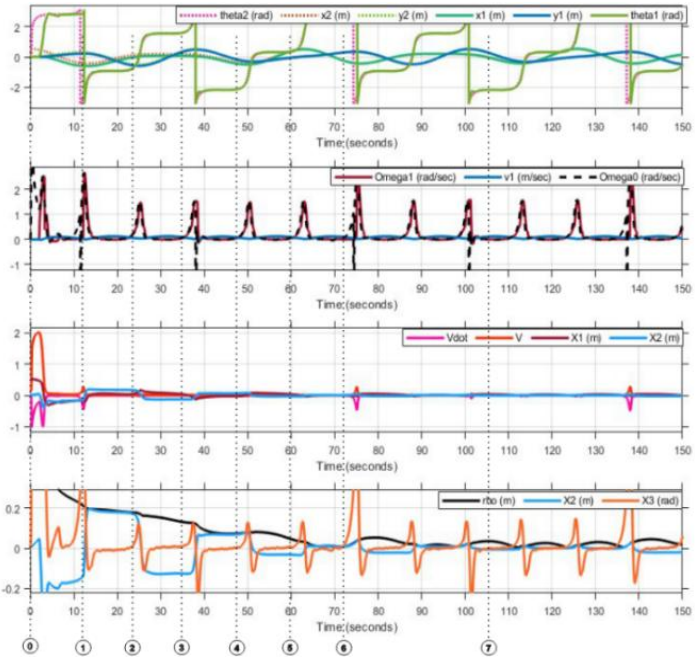

Figure 6. Hypotrochoid trajectory tracking: time plots of key parameters

From the second plots in Figure 6 we can also notice that the linear speed of the robot is virtually constant during several portions of the chosen trajectory. These correspond to the phases where the shape of the trajectory is close to a straight line, on the other hand this becomes very close to zero when the robot has to go through sharp V-turns, in which case its angular speed gets much higher. To avoid the undesirable sudden changes caused by the transition of θ_2 between $+\pi$ to $-\pi$, as at around $38\ sec$, the control signal Ω_1 is kept at the same value when $|\pi - \theta_2| < 0.3\ rad$, this issue has also been mentioned in [30], Ω_0 is the unwanted steering command which would have been applied without this limiting mechanism, in case which an overconsumption power would occur besides some expected mechanical damage of the motor reduction gears, caused by the sudden change of the rotational direction of the robot wheels. The set of plots shown on the third row of Figure 6 confirms the boundedness of the Lyapunov function V, its derivative $\dot{V}$ and the three systems states X_1, X_2, and X_3. A zoomed view of the Euclidean distance between the virtual and actual robot, designated ρ in Figure 1, together with the lateral distance X_2 and the orientation error X_3 is plotted on the lower part of Figure 6. This gives us more detailed information on the closeness between the actual robot and the desired trajectory. Considering the fact that the main goal of this work is to make the actual robot maintain its position on the desired trajectory and not necessarily very close to the virtual robot, the most important variable that should be looked at is the distance X_2. From the above plots we can notice that even if the actual robot is always more or less behind the virtual robot, the lateral distance X_2 is continuously decreasing, and has reached a value of less than 1.2 mm just before the end of this sample test.

4. CONCLUSION

The main focus of this paper is oriented towards the design and implementation of a nonlinear multivariable controller, specifically adapted to make a nonholonomic mobile robot follow a predefined trajectory. The results obtained from the first experimental test has shown that the robot was able to remain very close to the target trajectory, with a minor deviation occurrence after a sudden external disturbance purposely activated during the robot movements. Knowing that, from the theoretical point of view, this kind of mechanical perturbations is very difficult to prevent, the best method to assess the performance of such a controller is to rely on the acquired data measured from the actual hardware system. The second experimental test has shown that the time taken by the robot was quite longer compared to the previous test. This can be explained by the fact that during the initial transition, the controller action has given a higher priority to the orientation steering part, driving the robot trajectory to be parallel to the reference path. With the help of some optimization techniques the kinematic controller coefficients K_1 and K_1 could probably enhance the performance of the proposed controller.

FUNDING INFORMATION

This work is partially supported by the Algerian Ministry of Higher Education and Scientific Research through the PRFU project no. A01L08UN090120220001.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Boualem Kazed	✓	✓	✓	✓	✓	✓		✓	✓	✓			✓	
Abderrezak Guessoum		✓				✓		✓	✓	✓	✓	✓		

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest relevant to this study.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author, [BK], upon reasonable request.

REFERENCES

[1] Y. Kanayama, Y. Kimura, F. Miyazaki, and T. Noguchi, "A stable tracking control method for an autonomous mobile robot," in *Proceedings of the 1990 IEEE International Conference on Robotics and Automation*, May 1990, vol. 381, pp. 384–389, doi: 10.1109/robot.1990.126006.

[2] 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, 2020, doi: 10.35840/2631-5106/4125.

[3] S. Yigit and A. Sezgin, "Trajectory tracking via backstepping controller with PID or SMC for mobile robots," *Sakarya University Journal of Science*, vol. 27, no. 1, pp. 120–134, 2023, doi: 10.16984/saufenbilder.1148158.

[4] X. Gao, L. Yan, and C. Gerada, "Modeling and analysis in trajectory tracking control for wheeled mobile robots with wheel skidding and slipping: Disturbance rejection perspective," *Actuators*, vol. 10, no. 9, 2021, doi: 10.3390/act10090222.

[5] P. Petrov and I. Kralov, "Exponential trajectory tracking control of nonholonomic wheeled mobile robots," *Mathematics*, vol. 13, no. 1, 2025, doi: 10.3390/math13010001.

[6] E. J. Hwang, H. S. Kang, C. H. Hyun, and M. Park, "Robust backstepping control based on a Lyapunov redesign for Skid-Steered Wheeled Mobile Robots," *International Journal of Advanced Robotic Systems*, vol. 10, no. 1, 2013, doi: 10.5772/55059.

[7] H. Aithal and S. Janardhanan, "Trajectory tracking of two wheeled mobile robot using higher order sliding mode control," in *2013 International Conference on Control, Computing, Communication and Materials, ICCCCM 2013*, 2013, pp. 1–4, doi: 10.1109/ICCCCM.2013.6648908.

[8] Z. Cao, Y. Zhao, and Q. Wu, "Adaptive trajectory tracking control for a nonholonomic mobile robot," *Chinese Journal of Mechanical Engineering (English Edition)*, vol. 24, no. 4, pp. 546–552, 2011, doi: 10.3901/CJME.2011.04.546.

[9] Y. Koubaa, M. Boukattaya, and T. Damak, "Adaptive sliding-mode control of nonholonomic wheeled mobile robot," in *STA 2014 - 15th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering*, 2014, pp. 336–342, doi: 10.1109/STA.2014.7086759.

[10] C. Y. Chen, T. H. S. Li, Y. C. Yeh, and C. C. Chang, "Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots," *Mechatronics*, vol. 19, no. 2, pp. 156–166, 2009, doi: 10.1016/j.mechatronics.2008.09.004.

[11] S. Fadlo, A. A. Elmahjoub, and N. Rabbah, "Optimal trajectory tracking control for a wheeled mobile robot using backstepping technique," *International Journal of Electrical and Computer Engineering*, vol. 12, no. 6, pp. 5979–5987, 2022, doi: 10.11591/ijece.v12i6.pp5979-5987.

[12] 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.

[13] 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.

[14] M. G. Abdalnasser, M. Elsamanty, and A. A. Ibrahim, "Trajectory tracking of wheeled mobile robot through system identification and control using deep neural network," *Engineering Research Journal*, vol. 183, no. 3, pp. 1–17, 2024, doi: 10.21608/erj.2024.304953.1075.

[15] H. Huang and J. Gao, "Backstepping and novel sliding mode trajectory tracking controller for wheeled mobile robots," *Mathematics*, vol. 12, no. 10, p. 1458, 2024, doi: 10.3390/math12101458.

[16] Z. Anwar Ali, D. Wang, M. Safwan, W. Jiang, and M. Shafiq, "Trajectory tracking of a nonholonomic wheeled mobile robot using hybrid controller," *International Journal of Modeling and Optimization*, vol. 6, no. 3, pp. 136–141, 2016, doi:

10.7763/ijmo.2016.v6.518.
[17] J. Li, P. Miao, L. Shao, H. Liu, and X. Chen, "Trajectory tracking control of photovoltaic cleaning robot based on Lyapunov theory and Barbalat lemma," in *Proceedings of the 30th Chinese Control and Decision Conference, CCDC 2018*, 2018, pp. 2705–2709, doi: 10.1109/CCDC.2018.8407584.
[18] 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.
[19] T.-S. Jin and H.-H. Tack, "Path following control of mobile robot using Lyapunov techniques and PID controller," *International Journal of Fuzzy Logic and Intelligent Systems*, vol. 11, no. 1, pp. 49–53, 2011, doi: 10.5391/ijfis.2011.11.1.049.
[20] L. Yang and S. Pan, "A Sliding mode control method for trajectory tracking control of wheeled mobile robot," *Journal of Physics: Conference Series*, vol. 1074, p. 12059, 2018, doi: 10.1088/1742-6596/1074/1/01205.
[21] N. Uddin, "Trajectory tracking control system design for autonomous two-wheeled robot," *JURNAL INFOTEL*, vol. 10, no. 3, p. 90, 2018, doi: 10.20895/infotel.v10i3.393.
[22] E. Kayacan and G. Chowdhary, "Tracking error learning control for precise mobile robot path tracking in outdoor environment," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 95, no. 3–4, pp. 975–986, 2019, doi: 10.1007/s10846-018-0916-3.
[23] T. Beharu Arega, Y. Mersha Tesfa, and C. M. Abdissa, "Three-wheeled mobile robot trajectory tracking control using nonlinear PID controller based neural network combined with backstepping controller," *IEEE Access*, vol. 13, pp. 100167–100182, 2025, doi: 10.1109/ACCESS.2025.3577269.
[24] Y. A. Wendemagegn, W. Ayalew Asfaw, C. M. Abdissa, and L. N. Lemma, "Enhancing trajectory tracking accuracy in three-wheeled mobile robots using backstepping fuzzy sliding mode control," *Engineering Research Express*, vol. 6, no. 4, p. 45204, 2024, doi: 10.1088/2631-8695/ad79b9.
[25] C. A. Kedir and C. M. Abdissa, "PSO based linear parameter varying-model predictive control for trajectory tracking of autonomous vehicles," *Engineering Research Express*, vol. 6, no. 3, p. 35229, 2024, doi: 10.1088/2631-8695/ad722e.
[26] N. W. Madebo, C. M. Abdissa, and L. N. Lemma, "Enhanced trajectory control of quadrotor UAV using fuzzy PID based recurrent neural network controller," *IEEE Access*, vol. 12, pp. 190454–190469, 2024, doi: 10.1109/ACCESS.2024.3516494.
[27] E. A. Gedefaw, C. M. Abdissa, and L. N. Lemma, "An improved trajectory tracking control of quadcopter using a novel sliding mode control with fuzzy PID surface," *PLoS ONE*, vol. 19, no. 11 November, p. e0308997, 2024, doi: 10.1371/journal.pone.0308997.
[28] W. Ayalew, M. Menebo, C. Merga, and L. Negash, "Optimal path planning using bidirectional rapidly-exploring random tree star-dynamic window approach (BRRT*-DWA) with adaptive Monte Carlo localization (AMCL) for mobile robot," *Engineering Research Express*, vol. 6, no. 3, p. 35212, 2024, doi: 10.1088/2631-8695/ad61bd.
[29] K. J. Åström and B. Wittenmark, *Adaptive control*, 2nd ed. Addison-Wesley Longman Publishing Co., Inc, 1994.
[30] I. A. Hassan, I. A. Abed, and W. A. Al-Hussaibi, "Path planning and trajectory tracking control for two-wheel mobile robot," *Journal of Robotics and Control (JRC)*, vol. 5, no. 1, pp. 1–15, 2024, doi: 10.18196/jrc.v5i1.20489.

BIOGRAPHIES OF AUTHORS

Boualem Kazed received the Ing. d'etat degree in electronics engineering from the National Polytechnics of Algiers in 1985, an MPhil degree from the control engineering department of the university of Sheffield, UK, in 1988 and a PhD degree from the University of Blida, where he has been teaching different graduate and postgraduate courses in electronics and control engineering since 1989. Dr. Kazed 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 kazed_boualem@univ-blida.dz.

Abderrezak Guessoum received a BS degree in electronics from National Polytechnics of Algiers, in 1976. He then went to Georgia Institute of Technology in Atlanta, USA, where he obtained a master's in electrical engineering in 1979, a master's in applied mathematics in 1980 and 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, during 1984-1985. Since 1988, he has been with the Université de Blida, as a professor in the department of Electronics. Prof. Guessoum is the head of the Digital Signal and Image Processing Research Laboratory. His current research interests include digital and adaptive filter design, neural networks, genetic algorithms, robust and intelligent 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.