Journal of Applied Research and Technology

www.jart.icat.unam.mx

Journal of Applied Research and Technology 22 (2024) 510-517

Original

Control of the reference posture of a mobile robot with differential type wheels using the intermediate point technique

Ó. Gómez[a,b]* • J. Figueroa[a] • M. Montaño[a] • R. Rovira[a]

[a] *Universidad Estatal Península de Santa Elena, La Libertad, Ecuador*
[b] *Universidad Nacional de Colombia, Manizales, Colombia*

Received 01 22 2024; accepted 05 13 2024
Available 08 31 2024

Abstract: This manuscript aims to implement and evaluate a control approach known as a midpoint to guide a wheeled mobile robot toward a reference position and orientation. Instead of prescribing a specific path, the robot can choose any feasible way to reach the goal.

The method used uses a control algorithm based on the intermediate point. Initially, a controller based on the orientation error $e_{\varphi}(t)$and the distance to the target is used to guide the robot towards the desired reference point. The angular velocity $\omega(t)$ is controlled as a function of the orientation error $e_{\varphi}(t)$. In contrast, the translational velocity $v(\mathrm{t})$ is adjusted in proportion to the distance to the reference point. The kinematic model of the mobile robot with differential-type wheels is used to validate the algorithm.

The control algorithm based on the midpoint technique is verified using a kinematic model of a wheeled mobile robot. A reference posture $(x_{ref}, y_{ref}, \varphi_{ref})$ and an initial posture of the robot are established. The generated trajectory is verified in simulation using the MATLAB software and physically using the mobile robot "mbot neo" from the Makeblock company. After several tests, the results show that the robot has reached the specified reference position with the required orientation.

Keywords: Control, mobile robot, intermediate point, speed, odometry, posture

*Corresponding author.
E-mail address: oscargomez@upse.edu.ec (Ó. Gómez).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

Motion control of wheeled mobile robots in an obstacle-free environment can be performed by controlling the motion from some initial posture to some goal posture (classical control where the trajectory of the intermediate state is not prescribed) or by following the reference trajectory (Brockett, 1983). For wheeled mobile robots with non-holonomic constraints, controlling the reference posture is more complex than following the circuit connecting the initial pose and the goal posture (Bowling & Veloso, 1999).

A non-smooth or time-varying controller should be applied for successful reference posture or trajectory tracking control because the controlled system is non-linear and time-varying (Blažič, 2014).

When the robot moves, non-holonomic constraints must be considered so that its path cannot be arbitrary (Klančar & Škrjanc, 2007). Another reason favoring trajectory tracking control approaches is that robots are usually driven in environments with various constraints, obstacles and some demands that somehow define the desired path that takes the robot to the walking posture goal (Gu & Hu, 2002).

When controlling non-holonomic systems, the control action must be decomposed into the anticipatory (forward) control and feedback control parts (Cortés, 2023).

This is alternatively recognized as control with two degrees of freedom. The initiative-taking control component is computed based on the specified path, and the derived inputs can subsequently be supplied to the system to guide (trail) the specified path in the open loop (absent the feedback sensor). (Cortés, 2023).

Nevertheless, relying solely on feedforward control is not feasible, given its vulnerability to disturbances and inaccuracies in the initial state; therefore, it is necessary to include a feedback part (Cortés, 2023). Two-degree-of-freedom power is natural and is used when controlling non-holonomic mechanical systems.

Wheeled mobile robots are dynamic systems in which it is necessary to apply adequate torque to the wheels to obtain the desired movement of the platform. Therefore, motion control algorithms must consider the dynamic properties of the system. This problem is usually addressed using cascade control schemes with the external controller for speed control and internal torque control (force, motor current, etc.) (Mamblona, 2023).

The external controller determines the required system speeds to navigate to the reference posture or follow the reference trajectory (Mamblona, 2023). Simultaneously, the more rapid internal controller computes the necessary torques (force, motor current, etc.) to attain the system speeds dictated by the external controller.

The interior driver must be fast enough to make the additional phase delay introduced into the system manageable. In most of the available mobile robot platforms, the internal torque controller is already implemented in the forum, and the user only controls the desired speeds of the system by implementing the control, considering only the system's kinematics (Ollero & Amidi, 1991).

In the research work presented in Peng (2023), a controller based on the curve-following algorithm was developed as a solution to the problem that the trajectory of the wheeled mobile robot could not be monitored accurately. Firstly, using a global coordinate system, kinematic and dynamic models of the wheeled mobile robot are created. Then, the kinematic model of the actual and planned trajectories is used to determine the trajectory tracking error of the wheeled mobile robot. Based on the trajectory tracking error system model, the curve-following algorithm controller is employed in the outer loop control to eliminate the robot's position and attitude deviation during trajectory tracking, while a PI controller is used in the inner loop control to accurately track the robot's speed. Simulation results demonstrate that the suggested control technique can regulate the posture error of the wheeled mobile robot within ranges of 0.1 m, 0.03 m, and 0.03 rad, proving the efficacy of the control approach.

The paper given in Ce et al. (2018) presents an innovative method for trajectory tracking control in a non-holonomic wheeled mobile robot (NWMR), ensuring continuous robustness in finite time. The proposed strategy combines a conventional sliding mode controller (SMC) at the internal level with a modified switched second order sliding mode controller (S-SOSM) at the external level. The sliding mode controller is interpreted as a stabilization of the nominal system without uncertainties, while the S-SOSM algorithm is employed to counteract the effects of state-dependent unmodeled dynamics and time-varying external disturbances, significantly reducing the impact of unforeseen noise. The effectiveness and applicability of this approach are demonstrated through simulations and experiments.

In the work presented by the authors in Xu et al. (2022), a combined control scheme with backstepping and fractional-order PID for trajectory tracking of a differential drive mobile robot is developed. The kinematic and dynamic models of the mobile robot are described in detail for the design of the trajectory tracking controller. Then, based on the mobile robot model, the design of the trajectory tracking control system is addressed by combining backstepping with fractional-order PID. Moreover, to achieve an optimal control system, an enhanced particle swarm optimization algorithm is presented to adjust the parameters of the kinematic and dynamic controllers simultaneously.

The author of (Uddin, 2019), presents the design of a trajectory-tracking control system for a two-wheeled robot using the model reference adaptive control method. The controller allows the mobile robot to follow a desired trajectory. The tracking system dynamics are represented by posture error dynamics, which are derived from the robot's kinematics. Assuming small turning angles, the posture error dynamics are addressed through a linear system. The model reference adaptive control (MRAC) is applied in designing the controller based on the linear posture error dynamics. The performance of the controller was evaluated through computer simulation. The simulation results show that the designed controller can make the mobile robot follow a desired trajectory.

2. Materials and methods

This section details the materials and methods for executing this innovative approach. The methodology is based on a specific control algorithm, where an initial controller directs the robot towards the desired reference point, dynamically adjusting the angular and translational speed depending on the orientation error and the distance to the target.

The algorithm is validated using a kinematic model of the mobile robot, which provides a solid basis for understanding its operation in different scenarios. This study not only presents a transparent and replicable methodology for implementing the "middle point" approach but also highlights the robustness and effectiveness of the algorithm, verified both in simulation using MATLAB and in physical experiments with Makeblock's “mbot neo” robot.

2.1. Reference position control

This section explains some basic approaches to controlling a wheeled mobile robot to the reference posture, where the position and orientation define the reference posture. In this case, the route or trajectory to the reference posture is not prescribed (established), and the robot can drive along any feasible way to reach the goal. This path can be defined explicitly and adapted during movement or implicitly by performing the control algorithm applied to achieve the reference position.

Because only the initial (or current) posture and the final (or reference) posture are given, furthermore, the trajectory connecting these two positions is arbitrary, presenting new opportunities, such as opting for an "optimal" route. It is not necessary to emphasize that a feasible route must be selected where all restrictions, such as kinematics, dynamics, and those of the environment, are considered.

This still presents an endless array of potential pathways, from which a specific one is selected while adhering to additional criteria like time, length, curvature, energy consumption, and similar factors. In general, route planning poses a formidable challenge, and these aspects have not been considered in this study. (Yarad & Gómez, 2021).

Next, the reference posture control will be divided into two tasks: Control of orientation and control of forward motion constitute two fundamental components. These foundational elements are not deployable in isolation; however, when amalgamated, diverse control schemes emerge to attain the desired posture. These approaches are general and can be implemented in different kinematic models of mobile robots.

Therefore, this research focuses on implementing and evaluating an innovative control method called midpoint to guide a wheeled mobile robot to a specific position and orientation. Unlike conventional methods, this approach allows the robot to select flexible and viable routes to reach its destination. It seeks to highlight the relevance of this approach, outlining its theoretical foundations and the application of a control algorithm based on the intermediate point. Furthermore, experimental validation is anticipated using a kinematic model, evidencing the algorithm's effectiveness in simulation and the physical environment using the mbot neo mobile robot.

2.2. Orientation control

In every planar motion scenario, the need for orientation control arises to align with a specified orientation. The significance of orientation control is heightened by the non-holonomic constraints that limit the feasible directions of movement for a wheeled robot. While orientation control is inherently interdependent with forward motion control, the issue can also be examined through the lens of classical control to gain further understanding. This exploration will illustrate how control gains impact the classical closed-loop performance criteria of orientation control.

The orientation of the wheeled robot at some time t is $\varphi(t)$, and the reference or desired direction is $\varphi_{ref}(t)$.The control error can be defined as:

$$e_{\varphi}(t) = \varphi_{ref}(t) - \varphi(t) \tag{1}$$

Like any control system, a manipulating variable is essential to exert an impact on or alter the control variable, which in this context is the orientation. The objective of the control is to bring the control error to 0. In general, the convergence to 0 must be fast but respect additional requirements such as energy consumption, actuator load, system robustness in the presence of disturbances, noise, parasitic dynamics, etc.

The usual approach in control design is to start with the model of the system to be controlled. In this case, we work on the kinematic model, precisely its orientation equation.

2.3. Orientation control for a differential drive system

The kinematics of the differential drive system are expressed in Equation 2. The orientation equation of the kinematic model is as follows:

$$\dot{\varphi}(t) = \omega(t) \tag{2}$$

From the control point of view, Equation 2 describes the system with the control variable $\omega(t)$ and of integral nature (its pole is located at the origin of the complex plane s). It is widely recognized that a basic proportional controller has the capability to reduce the control error of an integral process to zero. The control law can be seen in Equation 3.

$$\omega(t) = Ke_{\varphi}(t) = K\left(\varphi_{ref}(t) - \varphi(t)\right) \tag{3}$$

Where the control gain K is an arbitrary positive constant, the interpretation of the control law is that the platform's angular velocity ω(t) is set proportionally to the orientation error of the robot. Combining Equations 1 and 2, the dynamics of the orientation control loop can be rewritten as:

$$\dot{\varphi}(t) = K\left(\varphi_{ref}(t) - \varphi(t)\right) \tag{4}$$

From where we can obtain the closed-loop transfer function of the controlled system, which is given by Equation 5.

$$G_{cl}(s) = \frac{\phi(s)}{\phi_{ref}(s)} = \frac{1}{\frac{1}{K}s+1} \tag{5}$$

With $\phi(s)$and $\phi_{ref}(s)$ being the Laplace transforms of $\varphi(t)$and $\varphi_{ref}(t)$, respectively. The transfer function $G_{cl}(s)$ is first order, which means that in the case of a constant reference, the orientation approaches the reference exponentially (with the time constant $\tau = 1/K$). The closed-loop transfer function also has unity gain, so there is no steady-state orientation error.

It has been shown that the controller makes the closed-loop transfer function behave like a first-order system. A second-order transfer function $G_{cl}(s) = \frac{\phi(s)}{\phi_{ref}(s)}$ is sometimes desired because it gives more degrees of freedom to the designer in terms of transient shape. The design was conducted by establishing the angular acceleration $\dot{\omega}(t)$ of the platform proportional to the orientation error of the robot:

$$\dot{\omega}(t) = K\left(\varphi_{ref}(t) - \varphi(t)\right) \tag{6}$$

With transfer function:

$$G_{cl}(s) = \frac{\phi(s)}{\phi_{ref}(s)} = \frac{K}{s^2+K} \tag{7}$$

Which corresponds to a second-order system with natural frequency $\omega_n = \sqrt{K}$ and damping coefficient ζ=0. Such a system is marginally stable, and its oscillatory responses are unacceptable. Damping is achieved by including an additional term to the controller, as shown in Equation 8.

$$\dot{\omega}(t) = K_1\left(\varphi_{ref}(t) - \varphi(t)\right) - K_2\dot{\varphi}(t) \tag{8}$$

Where K1 and K2 are arbitrary positive control gains, combining Equations 1 and 4, the closed loop transfer function becomes:

$$G_{cl}(s) = \frac{\phi(s)}{\phi_{ref}(s)} = \frac{K_1}{s^2+K_2s+K_1} \tag{9}$$

Where the natural frequency is $\omega_n = \sqrt{K_1}$, the damping coefficient $\zeta = K_2/2\sqrt{K_1}$ and the closed loop poles $s_{1,2} = -\zeta\omega_n \pm j\omega_n\sqrt{1-\zeta^2}$.

2.4. Forward motion control

Forward motion control refers to control algorithms defining the mobile robot's translational speed v(t) to achieve some control objective. Forward motion control alone cannot be used as a mobile robot control strategy. For example, suppose the desired orientation can be achieved by orientation control by taking the angular velocity $\omega(t)$ as the control signal in the case of differential drive. In that case, forward motion control alone cannot bring the robot to the desired position unless it is initially heading toward its target. This means that forward motion control is inevitably interconnected with orientation control.

However, the translation speed must be controlled to achieve the control objective. In the following trajectory, the rate is governed by the course, while in baseline posture control, the speed should decrease as we approach the final target. A reasonable idea is to apply control proportional to the distance to the reference point (x_{ref}, y_{ref}):

$$v(t) = K\sqrt{\left(x_{ref}(t) - x(t)\right)^2 + \left(y_{ref}(t) - y(t)\right)^2} \tag{10}$$

It must be considered that the reference position may be constant or change according to some reference trajectory. The control method does have certain limitations, particularly when dealing with extremely large or small distances to the reference point.

When the distance to the reference point is substantial, the control command provided by Equation 10 can become sizable. It is prudent to impose constraints on the maximum speed command. In practical applications, these constraints are often dictated by factors such as actuator limitations, driving surface conditions, trajectory curvature, and so forth.

If the distance to the reference point is minimal, the robot can effectively "overtake" the reference point (due to noise or imperfect vehicle model). The length increases as the robot moves away from the reference point and accelerates, according to Equation 10. This problem will be solved when forward motion controllers are combined with orientation controllers.

Velocity is intricately linked to acceleration, and in any practical implementation, the latter is constrained by the limited forces and torques applied by the actuators. This consideration holds significant importance when formulating forward motion control strategies. One possibility is to limit acceleration. It is sufficient to have a low pass filter at the output of the controller v(t) before the command is sent to the robot in the form of a signal $v^*(t)$. The simplest first-order filter with a DC gain of 1 can be used. A differential equation gives it:

$$\tau_f \dot{v}^*(t) + v^*(t) = v(t) \quad (11)$$

Or equivalent with a transfer function,

$$G_f(s) = \frac{V^*(s)}{V(s)} = \frac{1}{\tau_f s + 1} \quad (12)$$

where τ_f is a time constant of the filter.

2.5. Control to reference position

In this case, the robot must reach a reference (final) position where the absolute orientation is not prescribed (not set), which can be arbitrary. The robot's orientation is continuously controlled in the direction of the reference point to get to the reference point. This direction is denoted by $\varphi_r(t)$, as shown in Figure 1, which can be easily obtained using geometric relationships seen in Equation 13.

$$\varphi_r(t) = tan^{-1}\left(\frac{y_{ref} - y(t)}{x_{ref} - x(t)}\right) = artan\left(\frac{y_{ref} - y(t)}{x_{ref} - x(t)}\right) \quad (13)$$

Therefore, the angular velocity control $\omega(t)$ is ordered as follows:

$$\omega(t) = K_1 e_\varphi(t) = K_1(\varphi_r(t) - \varphi(t)) \quad (14)$$

Where K1 is a positive controller gain.

The translation speed of the robot is first ordered as proposed by the equation.

$$v(t) = K_2 D(t) = K_2 \sqrt{\left(x_{ref} - x(t)\right)^2 + \left(y_{ref} - y(t)\right)^2} \quad (15)$$

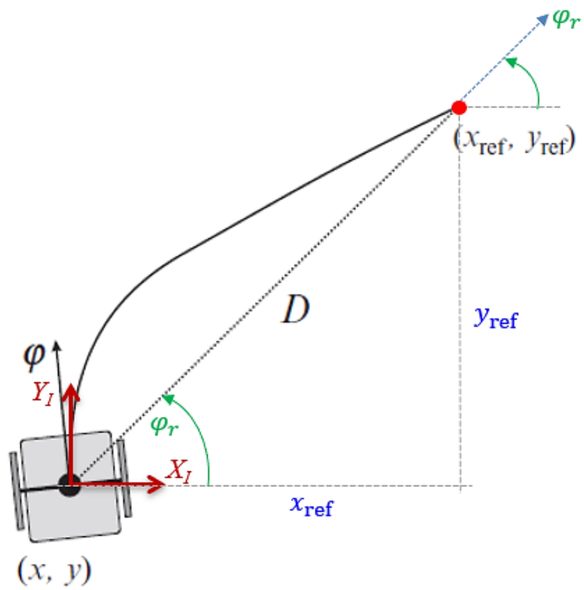

Figure 1. Control to the reference position.

The control law of Equation 15 also hides a potential danger when the robot approaches the target posture. The speed command is always positive, and when decelerating toward the final position, the robot can accidentally pass over it. The problem is that the speed command will increase after crossing the reference posture because the distance between the robot and the reference starts to grow. The other problem is that crossing the authority also causes the orientation of the connection to be opposite, which leads to rapid rotation of the robot. There are some simple solutions to this problem.

The orientation error changes abruptly when the robot passes over the reference point (±180 degrees). Therefore, the algorithm will check if the absolute value of the orientation error exceeds 90 degrees. The orientation error will increase or decrease by 180 degrees (so that it is in the range [-180 degrees, 180 degrees]) before it enters the controller. Furthermore, the control output of Equation 15 changes its sign in this case. Therefore, the mentioned problems are avoided by improved versions of the control laws.

$$\omega(t) = K_1 arctan\left(tan\left(e_\varphi(t)\right)\right) \quad (15)$$

$$v(t) = K_2 \sqrt{\left(x_{ref} - x(t)\right)^2 + \left(y_{ref} - yt(t)\right)^2} \;.\; sign\left(cos\left(e_\varphi(t)\right)\right) \quad (16)$$

The approach phase ends when the robot reaches a certain setpoint speed, and zero-speed commands are sent. This mechanism to stop the vehicle completely must be implemented even in the case of modified control law, especially in the case of noisy measurements.

2.6. Control to the reference posture using an intermediate point

This control algorithm is easy to implement because a simple controller given by Equations 14 and 15 is used, which drives the robot to the desired reference point. But in this case, not only the reference point $\left(x_{ref}(t), y_{ref}(t)\right)$ is required, but also the reference orientation φ_{ref} at the reference point. This approach aims to add an intermediate point that will model the trajectory so that the correct final orientation is obtained.

The intermediate point (x_t, y_t) is placed over the distance r from the reference point so that the direction from the middle point towards the reference point matches the reference orientation, as shown in Figure 2. The intermediate point is determined by Equation 17.

$$x_t = x_{ref} - r\ cos\varphi_{ref} \tag{17}$$

$$y_t = y_{ref} - r\ sin\varphi_{ref} \tag{18}$$

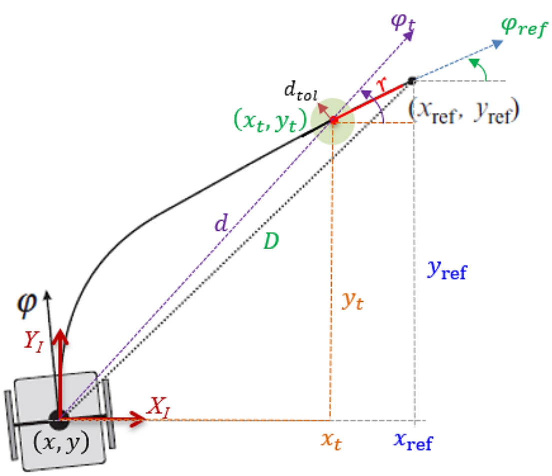

Figure 2. Reference posture control using an intermediate point.

The control algorithm consists of two phases. In the first phase, the robot is driven towards the intermediate point, considering that:

$$\varphi_t(t) = tan^{-1}\left(\frac{y_t - y(t)}{x_t - x(t)}\right) = artan\left(\frac{y_t - y(t)}{x_t - x(t)}\right) \tag{19}$$

$$e_\varphi(t) = \varphi_t(t) - \varphi(t) \tag{20}$$

When the distance to the intermediate point becomes low enough, verified by the condition $d < d_{tol}$ witch $d(t) = \sqrt{(x_t - x(t))^2 + (y_t - y(t))^2}$), The algorithm changes to the second phase, where the robot is controlled to the reference point using the following equation.

$$e_\varphi(t) = \varphi_{ref} - \varphi(t) \tag{21}$$

Table 1 presents the variation values obtained in various tests of the control algorithm. Parameters were modified, including the initial pose vector, reference position, and arrival orientation.

Table 1. Variation values of the control parameters for robot postures.

Starting posture vector	Reference position	Arrival orientation
q = [1; 0; -pi]	xyRef = [3; 2]	qRef = [3; 2; 3*pi/4]
q = [2; 1; -pi]	xyRef = [4; 3]	qRef = [4; 3; 4*pi/4]
q = [3; 2; -pi]	xyRef = [5; 3]	qRef = [5; 3; 5*pi/4]

Figure 3 shows the behavior of the linear velocity of the robot, which is given by m/s, vs. the angular momentum, which is provided by rad/s. When you have a value of angular velocity, the linear speed of the robot is zero.

Linear velocity increases proportionally to the product of angular momentum and radius of gyration, meaning that changes in angular velocity or radius of gyration will directly impact the linear rate of the robot. This relationship is crucial in the design and control of robots since it allows the resulting linear speed to be adjusted and predicted as a function of the angular velocity and the radius of gyration during different operations.

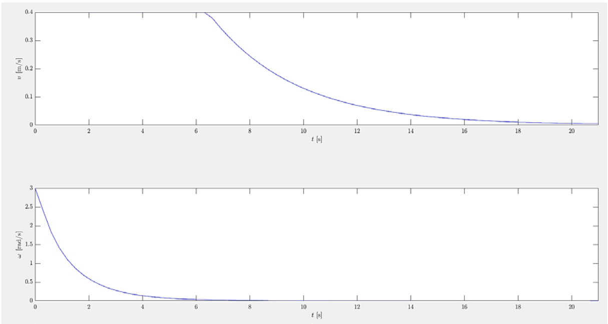

Figure 3. Results of the linear velocity and angular velocity of the robot.

During the execution of the reference posture control algorithm in the developed simulation, as shown in Figure 4, stable posture control is achieved with a minimum margin of error in the mean of ±0.02%. This result highlights the effectiveness and precision of the algorithm in managing the reference posture, evidencing a considerable level of accuracy and outstanding stability in tracking the desired position.

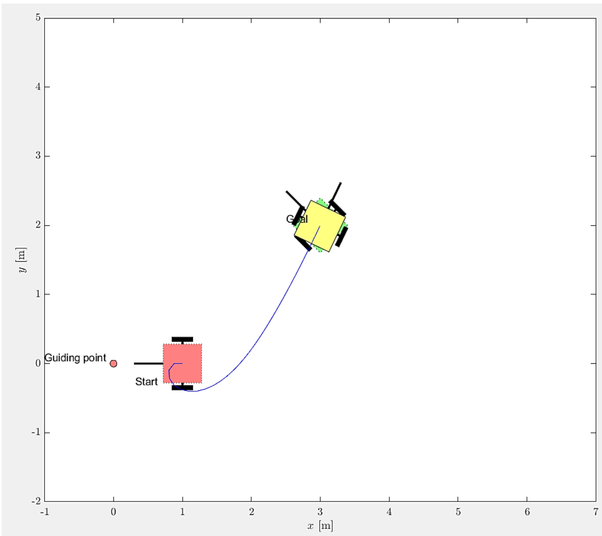

Figure 4. Simulation of the mobile robot posture with the midpoint algorithm.

To specify the linear velocity and angular velocity of the robot in the two phases, the following relationships are applied:

$$\omega(t) = K_1 e_{\varphi}(t) \tag{22}$$

$$v(t) = K_2 D(t) = K_2 \sqrt{\left(x_{ref} - x(t)\right)^2 + \left(y_{ref} - y(t)\right)^2} \tag{23}$$

This process guarantees that the robot reaches the reference position with the specified orientation; however, a slight error in the reference posture may arise. It is conceivable to introduce numerous variations to this algorithm, incorporating additional intermediate points, to enhance overall performance.

3. Conclusions

The mid-point technique for achieving a reference posture ensures that the wheeled mobile robot arrives at the reference position $\left(x_{ref}(t), y_{ref}(t)\right)$ with the required orientation φ_{ref}. Still, it may result in an error of orientation $e_{\varphi}(t)$ tiny in the reference pose.

It is possible to make many variations of this algorithm and make more intermediate points to improve performance. The algorithm described in this research work can be applied to many areas of mobile robotics, and depending on the application, the appropriate selection of the parameters r and d_{tol} must be made.

In the study, the route or trajectory to the reference posture has yet to be prescribed (established), and the robot can drive along any feasible way to reach the goal. This path is adapted and generated during the movement of the wheeled mobile robot according to the equations used by the control algorithm.

It is unnecessary to emphasize that a feasible route should be chosen where all constraints, such as kinematics, dynamics, and environment, are considered. This usually leaves an infinite selection of courses where a particular one is chosen, respecting additional criteria such as time, length, curvature, energy consumption, and the like. In general, route planning is challenging, and these aspects have yet to be considered in this research.

Conflict of interest

The authors have no conflict of interest to declare.

Funding

The authors received no specific funding for this work.

References

Brockett, R. W. (1983). Asymptotic stability and feedback stabilization. Differential geometric control theory, 27(1), 181-191.

Bowling, M., & Veloso, M. (1999). Motion control in dynamic multi-robot environments. In *Proceedings 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation*. (pp. 168-173). IEEE. https://doi.org/10.1109/CIRA.1999.810044

Blažič, S. (2014). On Periodic Control Laws for Mobile Robots. IEEE Transactions on Industrial Electronics, 7(61), 3660-3670. https://doi.org/10.1109/TIE.2013.2287222

Ce, H., Hongbin, W., Xiaoyan, C., Zhen, Z., Shungang, G., & Zhongquan, H. (2018). Finite-Time Switched Second-Order Sliding-Mode Control of Nonholonomic Wheeled Mobile Robot Systems. *Complexity*, *2018*(1), 1430989. https://doi.org/10.1155/2018/1430989

Cortés García, P. M., Mar Larios, R. E., Flores García, I. G., & Cruz López, E. (2023). Desarrollo de un prototipo de robot móvil autónomo con capacidad de desplazamiento y transporte de paquetes.

Gu, D., & Hu, H. (2002). Neural predictive control for a car-like mobile robot. *Robotics and Autonomous Systems*, *39*(2), 73-86. https://doi.org/10.1016/S0921-8890(02)00172-0

Klančar, G., & Škrjanc, I. (2007). Tracking-error model-based predictive control for mobile robots in real time. *Robotics and autonomous systems*, *55*(6), 460-469. https://doi.org/10.1016/j.robot.2007.01.002

Mamblona, C. J. (2023). Diseño del sistema de posicionamiento y percepción de un robot de competición mediante visión artificial y ROS. http://hdl.handle.net/10017/58114

Ollero, A., & Amidi, O. (1991). Predictive path tracking of mobile robots. Application to the CMU Navlab. In *Proceedings of 5th International Conference on Advanced Robotics, Robots in Unstructured Environments*, ICAR (Vol. 91, pp. 1081-1086).

Peng, Y., Zhang, P., Fang, Z., Zheng, S., & Guo, Z. (2023). Trajectory tracking control of the wheeled mobile robot based on the curve tracking algorithm. In *Journal of Physics: Conference Series* (Vol. 2419, No. 1, p. 012106). IOP Publishing. https://doi.org/10.1088/1742-6596/2419/1/012106

Uddin, N. (2019). Adaptive trajectory tracking control system of two-wheeled robot. In *Journal of Physics: Conference Series* (Vol. 1373, No. 1, p. 012052). IOP Publishing. https://doi.org/10.1088/1742-6596/1373/1/012052

Xu, L., Du, J., Song, B., & Cao, M. (2022). A combined backstepping and fractional-order PID controller to trajectory tracking of mobile robots. *Systems Science & Control Engineering*, *10*(1), 134-141. https://doi.org/10.1080/21642583.2022.2047125

Yarad, J. M. J., & Gómez, M. Ó. W. (2021). 6 DOF Automated Didactic Robotic Arm to Classify ballistic objects using artificial vision. *Journal of Security and Defense Sciences*, *6*(1), 14. https://doi.org/10.24133/RCSD.VOL06.N01.2021.04