Memorias del 2024 Congreso Nacional de Control Automático
8-11 de Octubre, 2024. Ciudad de México, México.


# Nonlinear Model Predictive Control for Pose Regulation Robot and Obstacle Avoidance ⋆

**Axel R. Rocha, Noé G. Aldana-Murillo** *,
**Edgar Martínez** **,***, **Emmanuel Ovalle-Magallanes** ****

* *Departamento de Ingenierías, Universidad Iberoamericana León, C.P. 37238, León, Guanajuato, México. (e-mail: 190084-7@iberoleon.edu.mx,noe.aldana@iberoleon.mx*
** *Consejo Nacional de Humanidades, Ciencias y Tecnologías, C.P. 03940, Ciudad de México, CDMX. (e-mail: edgar.martinez@conahcyt.mx)*
*** *Centro de Investigación en Matemáticas, C.P. 36023, Guanajuato, Gto., México. (e-mail: edgar.martinez@cimat.mx)*
**** *Dirección de Investigación y Doctorado, Facultad de Ingenierías y Tecnologías, Universidad La Salle Bajío, Av. Universidad 602. Col. Lomas del Campestre, León, 37150, Guanajuato, México., (e-mail: eovalle@lasallebajio.edu.mx)*

**Abstract:** The present work addressed the problem of autonomous navigation of wheeled mobile robots, specifically the Differential Driving Robot (DDR). The DDR kinematic model is nonlinear, which requires adequate automatic control strategies for good performance in autonomous navigation. The variables of the DDR mathematical model are the robot's position expressed in a global Cartesian reference frame, its orientation, its linear speed, and its angular speed. It was proposed that the navigation problem of a DDR be solved by following a reference trajectory using model-based nonlinear predictive control. In addition, a potential field algorithm was added for obstacle avoidance. Experiments were carried out in a dynamic simulator. Several simulations provided convincing evidence of the feasibility of implementing a real robot using the proposed approach.

*Keywords:* Autonomous Navigation, Differential Drive Robot, Nonlinear Model Predictive Control, Obstacle Avoidance.

## 1. INTRODUCTION

Nowadays, technology has focused on developing intelligent systems with different levels of autonomy to perform several tasks. For instance, there are multiple applications for robotics in industry and research areas, such as robotic arms in car manufacturing or performing highly accurate surgeries in medicine; mobile robots that transport products or people in indoor and outdoor environments, service robots for cleaning tasks, search and rescue robots in catastrophic circumstances and others.

In mobile robotics, executing a navigation policy is necessary so that the robot can travel between two different locations; this implies avoiding obstacles during the trajectory's execution. Multiple strategies for robotic navigation are reported in the literature; however, some of these approaches do not guarantee that the robot can solve these tasks by applying optimal velocities and generate discontinuities in the velocities of the robot, causing abrupt changes in their movements, inducing damage to the engines. Such as Sliding Modes Control (SMC), which is robust to external perturbations in the velocities; nevertheless, it has an oscillatory effect on the signal of the velocity of the robot, and it can generate physical damage to the engine of the robot over a long time of use (Jun and Lin, 2020; Zeng and Di, 2022; Li et al., 2022).

Although there are SMC schemes that can reduce the effect of chattering, the system's responses can be abrupt yet, compromising the robustness that characterizes these controllers. While the NMPC approach is more oriented toward local optimization, SMC tends to be more reactive, especially in applications with critical constraints. Additionally, the NMPC is a more flexible scheme capable of scaling to complex and nonlinear systems, integrating multiple objectives and constraints (Shtessel et al., 2023; Thi et al., 2023; Chawengkrittayanont et al., 2021). This motivates the research of autonomous navigation strategies and the development of more robust methodologies for collision avoidance in human environments. Therefore,

⋆ Supported by Universidad Iberoamericana León.

https://doi.org/10.58571/CNCA.AMCA.2024.042

implementing a control scheme that allows a soft variation of the robot's velocities is necessary. This way, physical damage to the robot and its engines or the environment resulting from its inertia for movement during obstacle evasion can be avoided. That is why it is crucial to predict the movement in the presence of an obstacle by controlling the robot's velocities. For these reasons, it is essential to use a Predictive Control Model (MPC) that, due to its predictive nature, allows the robot's speeds to be smoothed when detecting the obstacle to perform evasion subsequently. Additionally, it will enable the robot to avoid the obstacle while still moving toward the target. The MPC predicts the state over time, allowing the error in odometry measurements to be averaged, improving the precision of the robot's movement.

This manuscript proposes a nonlinear model predictive controller approach for autonomous navigation and obstacle avoidance of a differential drive robot (DDR). For that aim, we use the DDR's kinematics model. We use an artificial potential field algorithm for obstacle avoidance to generate virtual repulsive forces between the DDR and the obstacle. Our approach minimizes pose errors, optimizes control inputs, and achieves autonomous robot navigation.

The paper is organized as follows. Section 2 introduces the kinematic model of DDR; the obstacle avoidance strategy and the main problem are also presented. Section 3 contains the paper's main results. Here, we present a nonlinear model predictive controller in discrete time. This section is crucial to the paper, as it provides the methodology employed to develop the controller. Then, we present the simulation results in Section 4. Finally, Section 5 gives some conclusions and suggestions for future research.

### *1.1 Related work*

Extensive literature exists related to the problem of robot navigation. Some navigation methods focused on the issue of trajectory planning to avoid collisions with obstacles (Khatib, 1985; Borenstein and Koren, 1989; Minguez and Montano, 2004). Other schemes work particularly with nonholonomic robots, robots with movement restrictions, e.g., a car that can not move instantly in the perpendicular direction that its wheels are pointing (Bicchi et al., 1995; Hayet et al., 2014). In some other methods, a set of appropriate movement commands is determined to select a command based on a navigation strategy; for example, in (Martinez et al., 2019) presents an approach where a Differential Driving Robot (DDR), with a disc form, can explore unknown and connected environments with a scheme of hybrid navigation control that combines various feedback-based controllers with a states machine; besides, an algorithm called Gap Navigation Tree (GNT) is used to define if the environment has been explored during the navigation. The GNT is based on detecting discontinuities generated by corners in the environment.

Furthermore, in terms of optimal control, we have those that use MBP-based approaches. In (Bouzoualegh et al., 2018) presents a work to control a DDR using the dynamic

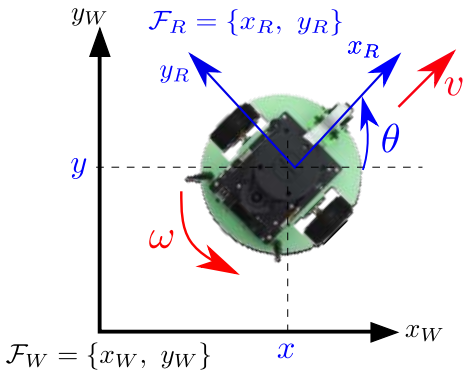


Fig. 1. Kinematic configuration of a differential drive robot. The world frame is denoted by $\mathcal{F}_W = \{x_W,\ y_W\}$, and the robot frame $\mathcal{F}_R = \{x_R,\ y_R\}$ is fixed to the body robot.

model of the robot. The model of the robot is linearized using an input-output technique, and based on this model, the necessary controls are obtained using MPC. They show the results of the work through simulations. The authors of (Lages and Vasconcelos Alves, 2006) proposed an optimal control for a DDR. The Taylor series linearized the mathematical model of the robot to make it possible to use predictive control based on the lineal model, solving the trajectory tracking problem of a DDR and achieving real-time implementation. In (Mehrez et al., 2015), nonlinear-MPC (NMPC) is used to compare two forms of restrictions of the optimization problem: equality and inequality constraints for trajectory tracking of a DDR. In (Hedjar, 2022) shows an NMPC for trajectory tracking of a mobile robot with wheels. They used an Euler method to transform the nonlinear optimization problem into a quadratic one. The results were presented in computer simulations and implemented in a real robot. The work presented in (Sani et al., 2021) shows an algorithm based in NMPC for autonomous navigation of a DDR where static and dynamic obstacles appear along its trajectory.

## 2. SYSTEM DESCRIPTION AND PROBLEM STATEMENT

This paper addresses the control problem of a DDR with a robust reactive static obstacle avoidance system. In this section, the kinematic, obstacle avoidance, and the problem formulation are presented.

### *2.1 Kinematic model of differential drive robots*

The kinematic model of a DDR (see Fig. 1) governs how wheel speeds map to robot velocities, such as:

$$\Sigma : \begin{cases} \dot{x} = v\cos\theta, \\ \dot{y} = v\sin\theta, \\ \dot{\theta} = \omega, \end{cases} \tag{1}$$

where the state variable $\mathbf{x} = [x,\ y,\ \theta]^\intercal$ denotes the position and the heading angle of the robot in the world frame $\mathcal{F}_W$. The control variable $\mathbf{u} = [v,\ \omega]^\intercal$ stands for the linear velocity and the angular velocity.

https://doi.org/10.58571/CNCA.AMCA.2024.042

*2.2 Obstacle avoidance using Artificial potential field algorithm*

A LIDAR (light detection and ranging) sensor detects an obstacle's position and size. The LIDAR detects the coordinates of the center of the obstacle $(x_{obs},\ y_{obs})$ in the frame $\mathcal{F}_R$, and the diameter of the obstacle $d_{obs}$.

The artificial potential fields approach was applied in the obstacle avoidance task. It consists of generating virtual forces from the center of the obstacle position to the outside, called repulsive forces. In this case, a function in which the repulsive force will increase exponentially as the distance between the object and the obstacle decreases was chosen. The next repulsive potential force was added:

$$F = \frac{F_{max}}{1 + e^{\alpha(\beta\left\|{}^{W}\mathbf{p}_{rob} - {}^{W}\mathbf{p}_{obs}\right\| - \gamma)}}, \tag{2}$$

where $F_{max}$ is a saturation on the force, $\alpha$, $\beta$, and $\gamma$ are gains that modify the behavior of the force curve. ${}^{W}\mathbf{p}_{rob} - {}^{W}\mathbf{p}_{obs}$ is the shortest vector from the center of the obstacle to the center of the robot.

*2.3 Problem formulation*

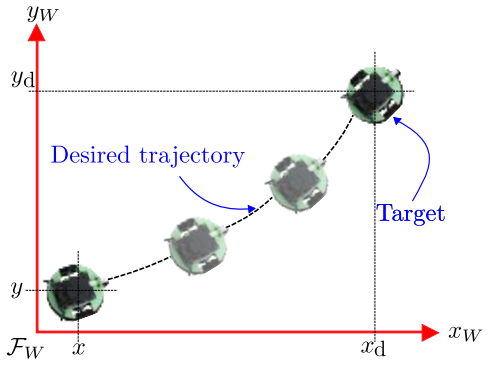


Fig. 2. This figure graphically defines an autonomous navigation problem. The robot goes from an initial pose $\mathbf{x} = [x,\ y]^\intercal$ to a target pose $\mathbf{x}_d = [x_d,\ y_d]^\intercal$ following a desired trajectory.

Consider the system described by Fig.2 with the model $\Sigma$ in (1). The available signals are the robot position $[x,\ y]^\intercal$, and the heading angle $\theta$. The problem involves finding a control law to bring state $\mathbf{x}$ to a user-predefined trajectory $\mathbf{x_d}$.

## 3. CONTROL

The main task of the NMPC is to find a control variable $\mathbf{u}$ to stabilize the system $\Sigma$ in (1). One must find a control $\mathbf{u}$ to achieve $\mathbf{x} \to \mathbf{x_d}$.

The NMPC can control a system that is typically a discrete-time model. At the same time that it controls the system, it satisfies a set of constraints on the states and controls. NMPC obtains a sequence of optimal controls by solving an optimization problem that minimizes an objective function. The first value of the controls obtained by solving the numerical optimization problem is applied to the system. Furthermore, an optimal trajectory of the controlled system can be predicted. The trajectory and the optimal controls are calculated numerically, considering the constraints given in a time window. The process is repeated from the system's current state to obtain a new sequence of controls and a new optimal trajectory. The system kinematics (1) have been discretized with the forward Euler method with sampling period $T_s = 0.1$ seconds.

Thus, the discrete NMPC can be formulated by solving the following constrained nonlinear optimization problem:

$$\begin{aligned} &\min_{x}\ J(x), \\ &\text{subject to} \\ &g_1(x) = c_1, \\ &g_2(x) \geq 0, \end{aligned} \tag{3}$$

where $J(x)$ is the cost function, $g_1(x) = c_1$ is a hard equality constraint, and $g_2(x) \geq 0$ is an inequality constraint. In this case, NMPC uses a model of the system being controlled, given by (1), to predict and optimize the control signals in some prediction window called the *horizon*. At each sampling time $k$, involves minimizing the cost function concerning the control sequence $\mathbf{u}(k) = [u(k), u(k+1), \cdots, u(k+n_u-1)]^\intercal$, where $n_u \geq 1$ is the control horizon.

The cost function to minimize with respect to $\mathbf{u}(k)$ is:

$$J = \sum_{i=0}^{n_u-1} \left\|\mathbf{u}([k+1]+i)\right\|_R^2 + \sum_{i=1}^{n_y} \left\|\mathbf{x}(k+i) - \mathbf{x_d}(k+i)\right\|_Q^2 \tag{4}$$

$$+ \sum_{i=1}^{n_y} \frac{F_{max}}{1 + e^{\alpha\left(\beta\left\|{}^{W}\mathbf{p}_{rob}(k+i) - {}^{W}\mathbf{p}_{obs}\right\| - \gamma\right)}}, \tag{5}$$

where $n_y \geq 1$ is the prediction horizon; $\mathbf{x}(k+i)$ are the predicted states; $\mathbf{u}([k+1]+i)$ are the predictive controls; ${}^{W}\mathbf{p}_{rob}(k+i)$ are the predicted robot´s positions in the frame $\mathcal{F}_W$; and $\mathbf{x_d}(k+i)$ denotes the reference state. The term $\|\cdot\|_W^2$ denotes the square norm of a vector weighted by a matrix $W$. In (4) $R \in \mathbb{R}^{2\times2}$, and $Q \in \mathbb{R}^{3\times3}$ are diagonal matrices. Notice that (5) is only used when the LIDAR detects an obstacle. The cost function (4) is subject to the hard equality constraints represented by the discretized system kinematics (1), i.e.,

$$\mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)), \tag{6}$$

where $\mathbf{x}(k) = [x(k),\ y(k),\ \theta(k)]^\intercal$. The inequality constraints correspond to the min and max controls

$$\mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max}. \tag{7}$$

The NMPC finds the optimal sequence of controllers $\mathbf{u}(k)$ by solving the optimization problem:

$$\min_{\mathbf{u}(k)}\ J\left(\mathbf{u}(k), \mathbf{x}(k), {}^{W}\mathbf{p}_{obs}, n_y, n_u\right), \tag{8}$$

$$\text{subject to:} \begin{cases} \mathbf{x}(k+1) = f(\mathbf{x}(k), \mathbf{u}(k)), \\ \mathbf{u}_{min} \leq \mathbf{u}(k) \leq \mathbf{u}_{max}. \end{cases}$$

which leads to the optimal sequence of control inputs $\mathbf{u}(k)$. Typically, the system only applies the first element $u(k)$ from the control sequence. The minimization

 

Table 1. Simulation parameters.

| Parameter | Description | Values |
|---|---|---|
| $T_s$ | Sampling time | 0.1 |
| R | Controls weight matrix | diag(0.002, 0.002) |
| Q | States weight matrix | diag(15.0, 15.0, 0.1) |
| $n_y$ | Prediction horizon | 15 |
| $n_u$ | Control horizon | 15 |
| $F_{max}$ | Force of the saturation | 20 |
| $\alpha$ | Gain of the force curve | 100 |
| $\beta$ | Gain of the force curve | 0.5 |
| $\gamma$ | Gain of the force curve | 1 |

problem is then reinitialized in the subsequent iteration, leading to the computation of a new sequence of optimal control inputs. This loop continues until the desired objective is accomplished.

## 4. SIMULATION RESULTS

The results presented in this section are divided into two parts using two obstacles. First, simulation results are presented only using PYTHON. The second part shows results obtained with two obstacles using ROS in its Python version and GAZEBO simulator. In both experiments, we have also used CASADI (Andersson et al., 2019) library as a solver for the optimization problem. This is an open-source tool for nonlinear optimization. The experiments run on an *Intel 2.70 GHz i5 Core* processor. We have considered the physical parameters of a Tiago++ (Pal Robotics, Pujades, Barcelona, Spain). The initial pose of the Robot in the motion plane is always taken as $\mathbf{q}_0 = [0\text{m}, 0\text{m}, 0\,\text{deg}]^\intercal$, while the final pose is denoted by $\mathbf{q}_d$. Table 1 presents the manually tuned parameters selected for the controller simulation.

We implemented the method on a Jetbot platform equipped with a Jetson Nano board. However, the Jetbot's odometry system showed limitations in accurately capturing the robot's pose data. Therefore, we plan to implement the method using a motion capture (MOCAP) system to improve pose estimation accuracy. Due to this, we only present the results in simulations on PYTHON and ROS.

Fig. 3 shows an experiment to reach the desired pose $\mathbf{q}_d = [4\text{m}, 4\text{m}, 0.78525\text{rad}]^\intercal$ on the top-left of the figure obstacles, altogether with the desired trajectory and the real trajectory, are shown in the $x, y$ plane. $\mathbf{u}_{min} = [-0.75,\ -0.981747]^\intercal$ and $\mathbf{u}_{max} = [0.75,\ 0.981747]^\intercal$. The diameter of the obstacles is 0.38m, and their position is at $(1.5\text{m}, 1.4\text{m})$ and $(3.0\text{m}, 2.5\text{m})$. The desired pose is reached with good final accuracy in position and orientation around 9 *seconds*. As can be seen, a smooth trajectory for the pose is obtained despite the obstacle avoidance. In the remaining sub-figures, the evolution of the $x$-coordinate, $y$-coordinate, and *yaw* orientation converges to the target. Fig. 4 presents the evolution of the robot's velocities and pose errors (right). On the left, we can observe the variations in angular velocity as the robot avoids the obstacles. The linear velocity remains constant until the robot reaches the target.

In Fig. 5, the results of an experiment to reach the desired pose $\mathbf{q}_d = [3\text{m}, 3\text{m}, 0.78525\text{rad}]^\intercal$. $\mathbf{u}_{min} = [-0.6,\ -0.981747]^\intercal$ and $\mathbf{u}_{max} = [0.6,\ 0.981747]^\intercal$. The diameter of the obstacles is 0.42m, and their position are at $(1.5\text{m}, 1.5\text{m})$ and $(2.5\text{m}, 0.5\text{m})$. The trajectory of the robot is depicted at the top of the figure. As can be seen, the goal was reached with good accuracy, and the robot navigated between the two obstacles. The evolution of the pose errors converges to zero around 6.5*seconds*; see left-bottom of the figure. The controls can be seen in the right-bottom of the figure, and it can observe the abrupt variation of the angular velocity when the robot passes through the obstacles.

## 5. CONCLUSION

This work presented the autonomous navigation of a differential drive robot and obstacle avoidance employing only the LIDAR sensor mounted in the robot. Furthermore, designing a robust obstacle avoidance control for the robot using the nonlinear model predictive control, particularly an artificial potential field algorithm, was successfully implemented. The simulation results prove that the proposed nonlinear model predictive control accurately tracks the desired pose at low convergence rates.

Future work includes implementing the navigation algorithm in a real robot and developing a visual model predictive control that does not require the robot's odometry and uses only a single monocular camera mounted on the robot without compromising the accuracy and time of the model.

## ACKNOWLEDGEMENTS

The first author acknowledges the financial support of the Universidad Iberoamericana León.

## REFERENCES

Andersson, J.A.E., Gillis, J., Horn, G., Rawlings, J.B., and Diehl, M. (2019). CasADi – A software framework for nonlinear optimization and optimal control. *Mathematical Programming Computation*, 11(1), 1–36. doi:10.1007/s12532-018-0139-4.

Bicchi, A., Casalino, G., and Santilli, C. (1995). Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles. In *IEEE International Conference on Robotics and Automation*, volume 2, 1349–1354 vol.2. doi:10.1109/ROBOT.1995.525466.

Borenstein, J. and Koren, Y. (1989). Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(5), 1179–1187. doi:10.1109/21.44033.

Bouzoualegh, S., Guechi, E.H., and Kelaiaia, R. (2018). Model predictive control of a differential-drive mobile robot. *Acta Universitatis Sapientiae, Electrical and Mechanical Engineering*, 10(1), 20–41. doi:doi:10.2478/auseme-2018-0002.

Chawengkrittayanont, P., Pukdeboon, C., Kuntanapreeda, S., and Moore, E.J. (2021). Smooth second-order sliding mode controller for multivariable mechanical systems. *SN Applied Sciences*, 3, 1–18.

 

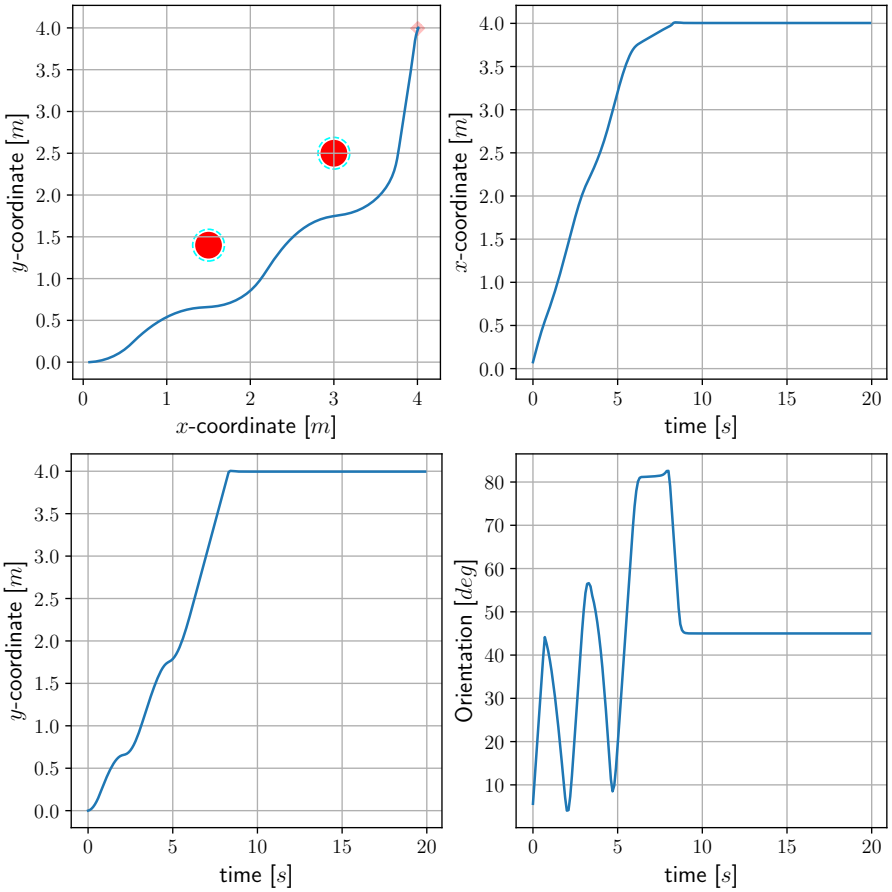


Fig. 3. Simulation of navigation using NMPC in the presence of two obstacles. In the figure in the top-left, the red circles are the obstacles, the dotted lines give the reference trajectory without obstacles, and the blue line is the trajectory executed by the robot. The evolution of the x-coordinate is shown in the top-right. In the bottom-left, the evolution of the y-coordinate is presented. The evolution of the orientation in degrees can be observed in the bottom-right.

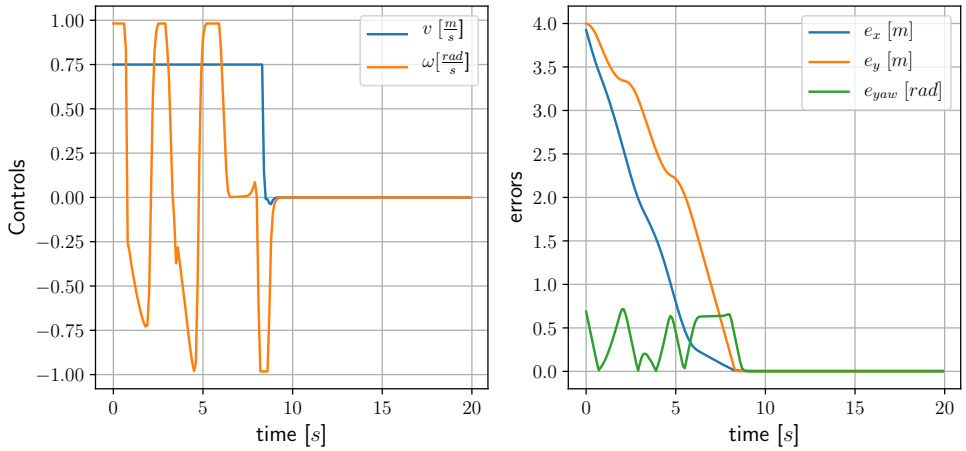


Fig. 4. Controls and errors obtained during navigation simulation using NMPC in the presence of two obstacles. On the left is the evolution of velocities. On the right is the evolution of pose errors to zero.

Hayet, J.B., Carlos, H., Esteves, C., and Murrieta-Cid, R. (2014). Motion planning for maintaining landmarks visibility with a differential drive robot. *Robotics and Autonomous Systems*, 62(4), 456–473. doi:https://doi.org/10.1016/j.robot.2013.12.003.

Hedjar, R. (2022). Approximate quadratic programming algorithm for nonlinear model predictive tracking control of a wheeled mobile robot. *IEEE Access*, 10,

 

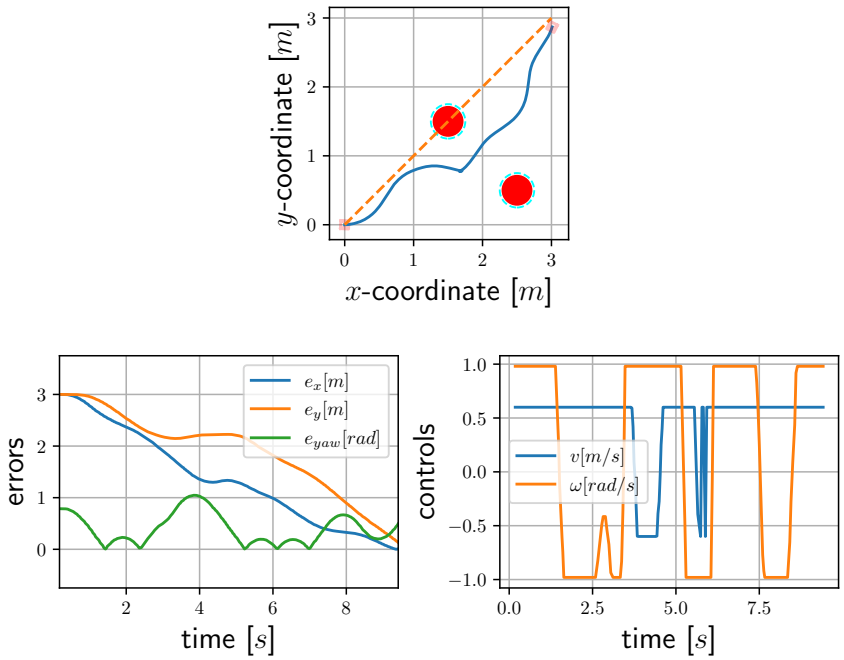


Fig. 5. Simulation of navigation using NMPC in the presence of two obstacles. In the figure in the top row, the two red circles represent the obstacles, the dotted lines give the reference trajectory without obstacles, and the blue line is the trajectory executed by the robot.

65067–65079. doi:10.1109/ACCESS.2022.3178727.

Jun, C. and Lin, W. (2020). Track tracking of double joint robot based on sliding mode control. In *IEEE International Conference on Information Systems and Computer Aided Education*, 626–631. doi:10.1109/ICISCAE51034.2020.9236895.

Khatib, O. (1985). Real-time obstacle avoidance for manipulators and mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 2, 500–505. doi:10.1109/ROBOT.1985.1087247.

Lages, W.F. and Vasconcelos Alves, J.A. (2006). Real-time control of a mobile robot using linearized model predictive control. *IFAC Proceedings Volumes*, 39(16), 968–973. doi:https://doi.org/10.3182/20060912-3-DE-2911.00166.

Li, J., Wang, J., Peng, H., Hu, Y., and Su, H. (2022). Fuzzy-torque approximation-enhanced sliding mode control for lateral stability of mobile robot. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 52(4), 2491–2500. doi:10.1109/TSMC.2021.3050616.

Martinez, E., Laguna, G., Murrieta-Cid, R., Becerra, H.M., Lopez-Padilla, R., and LaValle, S.M. (2019). A motion strategy for exploration driven by an automaton activating feedback-based controllers. *Autonomous Robots*, 43(7), 1801–1825. doi:10.1007/s10514-019-09835-6.

Mehrez, M.W., Mann, G.K.I., and Gosine, R.G. (2015). Comparison of stabilizing nmpc designs for wheeled mobile robots: An experimental study. In *Moratuwa Engineering Research Conference*, 130–135. doi:10.1109/MERCon.2015.7112333.

Minguez, J. and Montano, L. (2004). Nearness diagram (nd) navigation: collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1), 45–59. doi:10.1109/TRA.2003.820849.

Sani, M., Robu, B., and Hably, A. (2021). Dynamic obstacles avoidance using nonlinear model predictive control. In *47th Annual Conference of the IEEE Industrial Electronics Society*, 1–6. doi:10.1109/IECON48115.2021.9589658.

Shtessel, Y., Plestan, F., Edwards, C., and Levant, A. (2023). Adaptive sliding mode and higher order sliding-mode control techniques with applications: A survey. In *Sliding-Mode Control and Variable-Structure Systems: The State of the Art*, 267–305. Springer.

Thi, L.T., Quoc, H.P., Xuan, T.N., Dang, T.N., Quy, T.D., and Minh, D.D. (2023). Adaptive sliding mode control for series elastic actuator robots. In *International Conference on Engineering Research and Applications*, 159–168. Springer.

Zeng, H. and Di, Y. (2022). Trajectory tracking control of mobile robot based on interval type ii fuzzy sliding mode. In *IEEE International Conference on Electronic Technology, Communication and Information*, 1420–1425. doi:10.1109/ICETCI55101.2022.9832266.

 