DINAME 2023 - Proceedings of the XIX International Symposium on Dynamic Problems of Mechanics,
A.S. De Paula, A.T. Fabro and M.R. Machado (Editors), ABCM.
Pirenópolis, Go, Brazil, February 26th to March 3rd, 2023

# Cascade Control for Path Tracking of Autonomous Robotic Vehicles

**Vinicius Ribeiro Cavaleiro de Macedo**[1], **Marcelo Alejandro Forte**[2], **Gustavo Simão Rodrigues**[3], **Elias Dias Rossi Lopes**[4]

Instituto Militar de Engenharia
Praça General Tibúrcio, 80, 22290-270
Rio de Janeiro, RJ, Brasil
[1]vinicius.macedo@ime.eb.br
[2]mafroxden@ime.eb.br
[3]simao@ime.eb.br
[4]eliasrossi@ime.eb.br

*Abstract: One of the great challenges, nowadays, to make robotic systems, wheeled or tracked, fully functional is the optimization of the onboard control systems. Thus, the present work aims to study, computationally implement and analyze a cascade framework using a combination of Model Predictive Control (MPC) and a classical PID, for path tracking of an autonomous robotic vehicle. The MPC is used to define the forces on the body frame that are necessary for the path tracking, while the PID is used on the low-level control, tracking the reference forces defined on the first one. The dynamic model of a skid-steering vehicle is developed, considering friction efforts on the tire-road interaction, and the controlled model is simulated. The proposed control is proved to be efficient for path tracking, with proper mean square errors and processing time, which proves that the advantages of both techniques are associated on the cascade framework.*

***Keywords: MPC, cascade control, robotic vehicles, autonomous navigation***

## INTRODUCTION

Nowadays, the use of unmanned vehicles has spread massively in social, commercial and productive areas. Every day, new research, developments and approaches are applied. So, there are many applications for that type of technology, as space exploration rovers, mining, uninterrupted operation in agriculture, etc (Alatise and Hancke, 2020). Another application that stands is for military tasks, as bomb deactivation, rescue, patrol and even combat (Ni *et al.*, 2021). In addition, the main issue in driving and performance in the mentioned vehicles is the configuration of their trajectory control system.

In this context, one of the biggest challenges to making robotic vehicle systems, wheeled or tracked, functional is the efficiency and accuracy of the embedded control system. Currently, there is a wide range of control systems and algorithms, and among them, one stands out, the Model-based Predictive Control (MPC), presented by Cui *et al.* (2018). As a recent and powerful technique, there are many possibilities for improvement, which is evaluated by some authors, as Wang *et al.* (2019), in order to make the method more efficient and viable for online applications.

Škrjanc and Klančar (2017) suggests a continuous MPC for trajectory tracking. The authors argue that in many situations it is not possible to obtain a uniform sampling for the control, due to imperfection in the sensors, poorly synchronized clocks, non-deterministic delays in the control and due to the unknown pre-processing time. Thus, due to the continuous-time approach, the application of the proposed MPC is not limited to sampling uniformity. With this, it is demonstrated that in general situations involving uniform samples in time, the continuous MPC has a similar performance to its discrete version, but the former has better results when not, in addition to showing better robustness in the design parameters of the control laws.

Negri *et al.* (2017) presents a non-linear MPC with fault tolerance characteristics to be used in a two-degree-of-freedom robotic manipulator. The MPC used is obtained by training a fully connected cascade artificial neural network. Two different fault-handling approaches were used: the robust strategy and the adaptive strategy. The first one has a simpler implementation compared to the adaptive one and presents faster and more stable results compared to the MPC without fault handling. In the second, there is a greater improvement in the performance of the controller, but with the disadvantage of having more parameters to calibrate.

Song *et al.* (2019) proposes a control method for time-varying trajectory followers. It is made for a two-dimensional vehicle and the bicycle model is used for dynamic modeling. This method considers the influence of the longitudinal speed of the robot and the stability of the robot on the curvature of the trajectory under complex driving conditions at low speeds.

Farag (2021) develops an MPC controller that allows maneuverability in complex trajectories for autonomous vehicles, discussing all the project details and implementation stages of the proposed algorithm. For that, as the input signal, we have the precise position of the vehicle, its instantaneous speed and the reference trajectory to be followed, and as outputs, we have the butterfly valve command and the steering command. The bicycle model was used for the dynamic modeling of the autonomous vehicle. According to the author, one of the main contributions of the article is the form of the objective function used in the controller, which has 9 different types of terms, each with a weight, taking into account different aspects. With this, it was possible to simulate the control of autonomous vehicles in complex trajectories and with precision.

## Objective

The present work aims at the study, computational implement and analysis of the trajectory control of robotic vehicles using a cascade framework with Model Predictive Control. The proposed control framework is evaluated for the autonomous navigation of a friction-based skid-steering robotic vehicle. The main purpose of this strategy is to associate the accuracy of MPC with the low processing times of simpler control laws, as PID.

# METHODOLOGY

## Vehicle Dynamics

Figure 1 presents the robot body frame coordinate system, according to ISO 4130, and the fixed coordinate system. It demonstrates also the main variables evaluated on the vehicle performance: longitudinal velocity ($v_x$), angular velocity ($\omega$), yaw angle ($\phi$), wheels angular velocity ($\omega_1$ and $\omega_2$), wheels radius and longitudinal distance ($r$ and $L$).

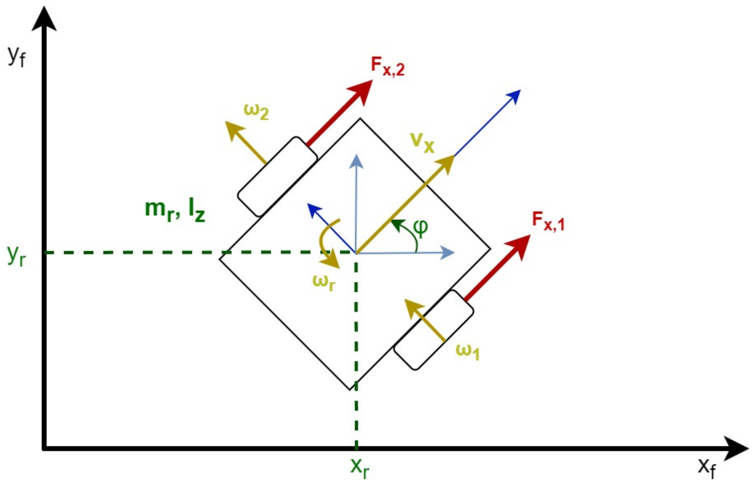


**Figure 1 – Reference System and Variables.**

As Wong (2022) in his simplified analysis of vehicle kinetics and dynamics in steering at low speed, where centrifugal force is neglected, vehicle behavior can be described by the following equations, Eq. (1) e Eq. (2), of motion:

$$m_r \frac{dv_x}{dt} = F_{x,1} + F_{x,2} \tag{1}$$

$$I_z \frac{d\omega_r}{dt} = F_{x,1} \cdot \frac{L}{2} - F_{x,2} \cdot \frac{L}{2} \tag{2}$$

Being $F_{x,1}$ and $F_{x,2}$ the longitudinal forces of the right wheels and left one, respectively, the product of the interaction between the weight of the vehicle ($F_z$) and the friction ($\mu$) of the soil in the longitudinal direction. Therefore, this will be the maximum longitudinal force that confers a movement without sliding and/or skidding, which is expressed as:

$$F_{x,max} = \mu F_z \tag{3}$$

So, it will be used a linearized form of the Coulomb friction (Piedbœuf *et al.*, 2000), as we consider that the wheels are rigid. In this formulation, the maximum value of the force occurs in a given sliding velocity between the wheel and the floor. It is assumed that the traction force varies with the sliding velocity as in Figure 2, where $v_{d,max}$ is the maximum sliding velocity. Mathematically, this curve can be described by the Equation 4.

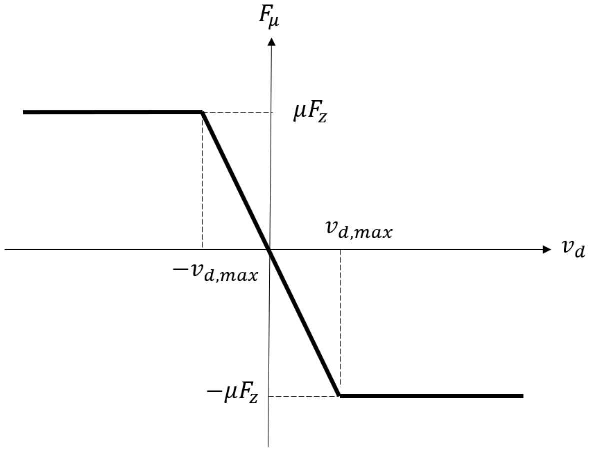


**Figure 2 – Linearized Coulomb friction.**

$$F_\mu = \begin{cases} -\mu F_z \cdot v_d / v_{d,max} & \text{, for } v_d \in (-v_{d,max}, +v_{d,max}) \\ -\mu F_z \cdot sign(v_d) & \text{, for } v_d \notin (-v_{d,max}, +v_{d,max}) \end{cases} \tag{4}$$

Still, using the kinematics of the system it is easy the relate the coordinates of the robot with its velocities, as shown in the following equation:

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} v_x \cdot cos(\phi) \\ v_x \cdot sin(\phi) \\ \omega_r \end{bmatrix} \tag{5}$$

For the dynamics of the wheel, using Euler's law, there is the following equation, which can be used for both wheels :

$$J_w \cdot \dot{\omega}_w = T - F_x \cdot r_w - c_v \cdot \omega_w \tag{6}$$

where $r_w$ is the radius of the wheel, $c_v$ is the viscous friction coefficient in the shafts, $T$ is the torques applied by the motor on the wheel.

## Model Predictive Control

The MPC is a control strategy based on an optimization process. This technique aims to define the control actions based on predicted future states or outputs of the system. The prediction is performed for a moving window of time with a predefined size, so that the dynamic model, physical constraints and future references are considered in the cost function. The basic structure of the MPC is presented in Figure 3. As an optimization problem, the control action is obtained through the minimization of the cost function, which, in this case, consider the error between the predicted outputs and references, and the control signal limits (Camacho and Alba, 2013).

## Proposed Control Strategy

For a cascade control, PID control is chosen to be used with the MPC. Figure 4 shows the strategy used. In this way, the states for the control of the MPC are defined by Equation 7, representing the dynamics of the robot chassis, returning the forces on the chassis as a reference input, as shown in Equation 8.

$$x_{mpc} = [x_r \quad y_r \quad \phi \quad v_x \quad \omega_r]^T \tag{7}$$

$$u_r = [F^r_{x,1} \quad F^r_{x,2}]^T \tag{8}$$

Then, the static relationship between the traction force and wheel slip is used to generate the secondary state values of reference, which are wheel slips, as shown in Equation 9.

$$o_r = [v^r_{d,1} \quad v^r_{d,2}]^T \tag{9}$$

Then, the reference slips are compared with the actual slips, shown in Equation 10, given by the plant output. These secondary states are calculated from the PID states, shown in Equation 11, and are given by Equations 12 and 13.

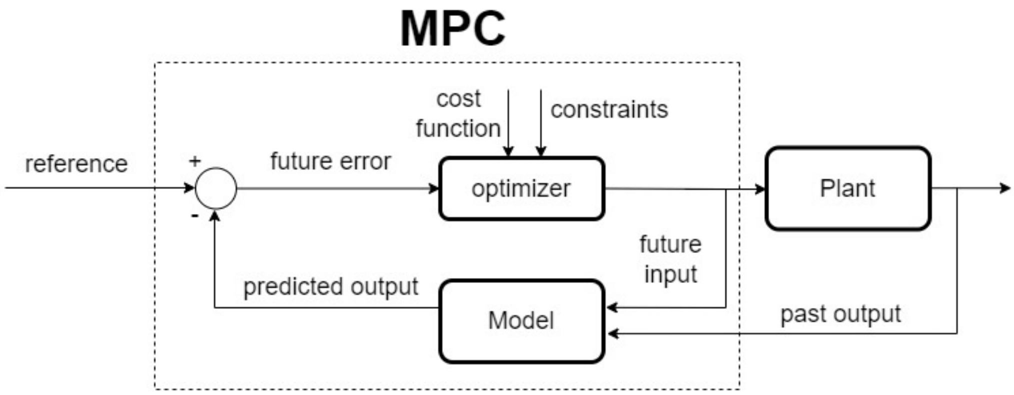


**Figure 3 – MPC strategy**

$$o = [v_{d,1} \quad v_{d,2}]^T \tag{10}$$

$$x_{pid} = [x_r \quad y_r \quad \phi \quad v_x \quad \omega_r \quad \omega_1 \quad \omega_2]^T \tag{11}$$

$$v_{d,1} = \left(v_x + \omega_r \frac{L}{2}\right) - \omega_1 \cdot r_w \tag{12}$$

$$v_{d,2} = \left(v_x - \omega_r \frac{L}{2}\right) - \omega_2 \cdot r_w \tag{13}$$

Finally, the PID calculates the plant control signal, given by Equation 14, and applies it to it, resulting in the outputs of Equation 15.

$$u = [T_1 \quad T_2]^T \tag{14}$$

$$z = [x_r \quad y_r \quad \phi \quad v_x \quad \omega_r]^T \tag{15}$$

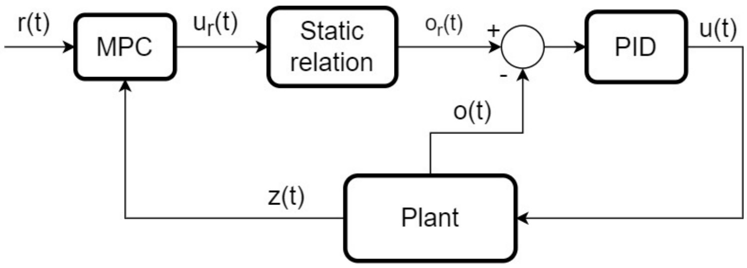


**Figure 4 – Cascade control strategy**

Its implementation can be done using MatLab software and the CasADi (Andersson *et al.*, 2019) library, which provides a highly flexible symbolic workspace for solving optimization problems that are constrained by differential equations.

## SIMULATION AND RESULTS

The dynamics of the rover shown in Figure 5 is mathematically modeled, using it as an object of analysis. Therefore, comparing the robot data with the skid-steering model shown in the vehicle dynamics, we have the physical parameters presented on Table 1. Furthermore, analyzing the physical limitations of the rover, and the consequent limits of the state variables, as presented on Table 2. To analyze the accuracy achieved by the control, the root mean square error (RMSE) is used, given by the Equation 16, where the subscript *i* refers to the iteration *i*, and *n* is the total number of iterations that were made in the simulation.

**Figure 5 – Rover Lynxmotion Aluminum 4WD1**

**Table 1 – Physical parameters**

| **Parameter** | **Symbol** | **Value** | **Unit** |
|---|---|---|---|
| Mass | $m_r$ | $4,0$ | $kg$ |
| Gravity | $g$ | $9,81$ | $N/kg$ |
| Rover moment of inertia | $I_z$ | $0,0338$ | $kg\cdot m^2$ |
| Distance between wheels | $L$ | $0,2735$ | $m$ |
| Tire soil friction coefficient | $\mu$ | $0,7$ | $-$ |
| wheel radius | $r_w$ | $0,06$ | $m$ |
| Polar moment of inertia of the wheel | $J_w$ | $3,31\cdot 10^{-5}$ | $kg\cdot m^2$ |
| Vicious friction coefficient | $c_v$ | $0,02$ | $Nms/rad$ |

**Table 2 – Physical limits for simulation**

| Entity | Minimum value | Maximum value |
|---|---|---|
| $v_x$ | $-0,9$ | $0,9$ |
| $\omega_r$ | $-9,1$ | $9,1$ |
| $\omega_1$ e $\omega_2$ | $-20,9$ | $20,9$ |
| $T_1$ e $T_2$ | $-0,45$ | $0,45$ |

$$RMSE_z = \sqrt{\sum_{i=1}^{n} \frac{(r(i) - z(i))^2}{n}} \tag{16}$$

After generating the RMSE of the outputs, this value is divided by a nominal value of the reference of that output to obtain a percentage notion of the error. Thus:

$$error_z(\%) = 100\frac{RMSEz}{z_{nominal}} \tag{17}$$

To choose the nominal values, the maximum values reached for each of the outputs were used, as shown in Table 3.

**Table 3 – Output nominal values**

| **Outputs** | $x_r$ | $y_r$ | $\phi$ | $v_x$ | $\omega_r$ |
|---|---|---|---|---|---|
| **Nominal values** | 1,0 | 1,0 | 1,571 | 0,524 | 1,0 |

In order to compare the accuracy and processing time of the proposed cascade control, the simulation of the complete MPC for the rover was performed, considering as input the desired path and as output the applied torque on the wheels. From this, the percentage error as defined was also calculated and is shown in Table 4.

Figures 6, 7 and 8 show the results for the cascade control strategy proposed in a trajectory with two different circular curves for a line change. It is possible to note that the tracking of the references is accurate, achieving precise results. Regarding the error, and initially analyzing the states $x_r$, $y_r$ and $\phi$, we notice losses of 0.19 , 0.13 and 0.19, respectively, on the error (%) value. It can be seen that this loss in precision for all states was less than one time the precision of the complete MPC, that is, in relation to the MPC the cascade control had an increase in the imprecision of less than 100%. However, it is notable that for $x_r$ this increase was tiny, only 0.19 compared to 2.20.

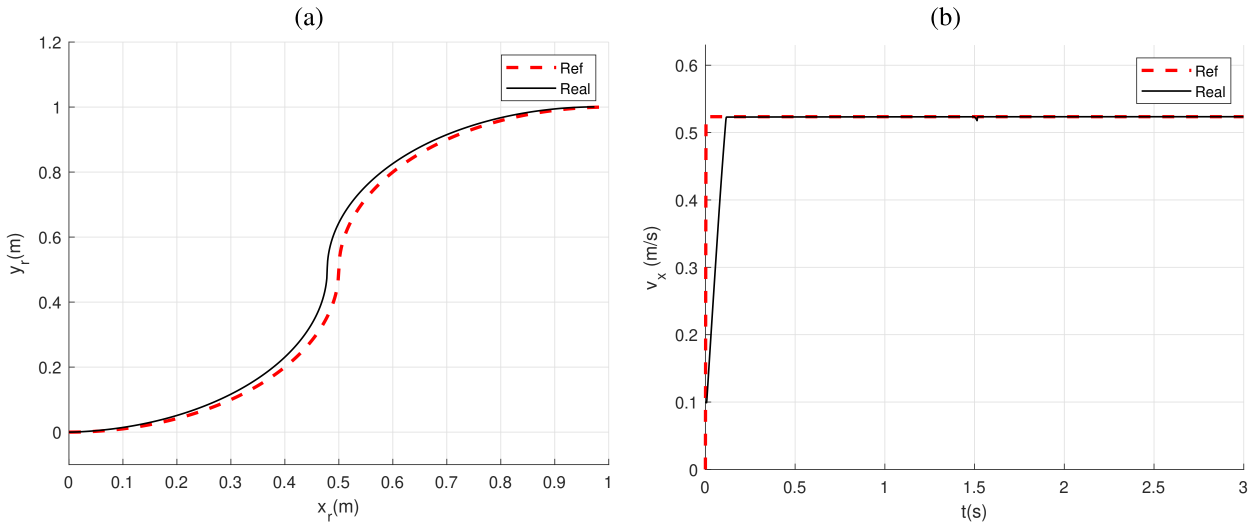


**Figure 6 – circular curves: (a) $x_r$ e $y_r$ (b) $v_x$**

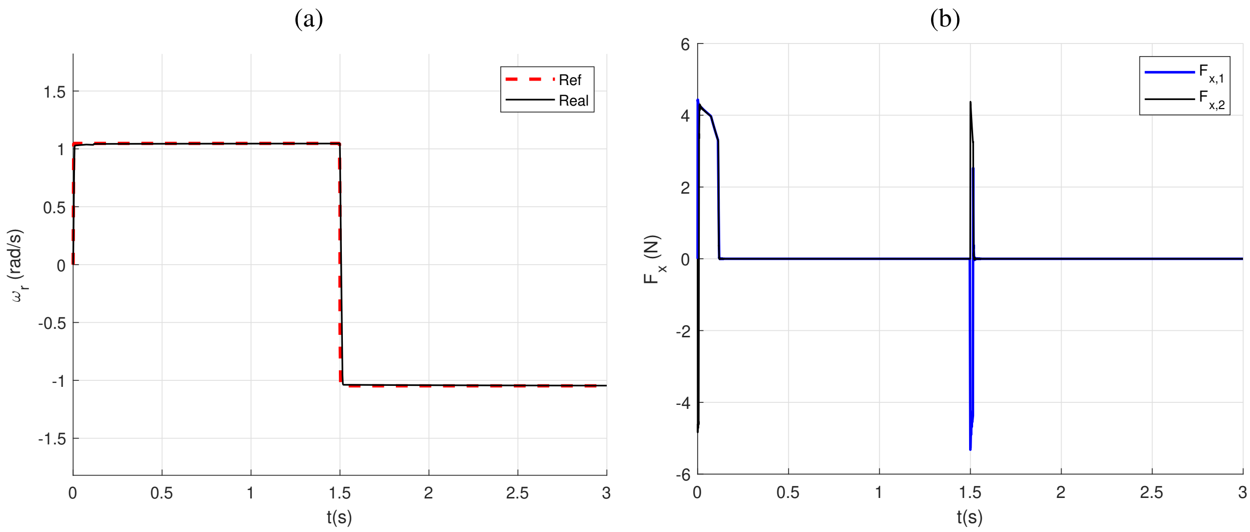


**Figure 7 – circular curves: (a) $\omega_r$ (b) $F_x$**

For $v_x$ and $\omega_r$ the default is maintained. There is an increase in the imprecision for each state by $9.50 - 8.85 = 0.65\%$ and $8.30 - 7.18 = 1.12\%$, respectively. These states are especially important regarding actual control implementations, as they are the commonly provided reference inputs for drivers and control boards. The use of references such as the position $x_r$ and $y_r$ is more difficult, as it requires more specific, costly, and difficult-to-implement sensing, such as GPS for online sensing. Thus, the low increase in imprecision gives good indications of functionality.

**Table 4 – Performance parameters**

| **Strategy** | **error(%)** $x_r$ | **error(%)** $y_r$ | **error(%)** $\phi$ | **error(%)** $v_x$ | **error(%)** $\omega_r$ | $\Delta t_m[s]$ |
|---|---|---|---|---|---|---|
| **Complete MPC** | 2,20 | 0,19 | 0,36 | 8,85 | 7,18 | 0,0193 |
| **Cascade Framework** | 2,39 | 0,32 | 0,55 | 9,50 | 8,30 | 0,0067 |

Additionally, analyzing the average processing time of an iteration of the two controllers, it is noted that the cascade strategy performed the processing in about 1/3 of the time required by the other controller. It should be noted that processing time is an important design requirement for online applications, in which the hardware is commonly limited, harming the effectiveness of the controller. In this way, the proposed cascade control is able to reduce the processing time by about a third at the cost of low follow-up imprecision, and its implementation was potentially successful.

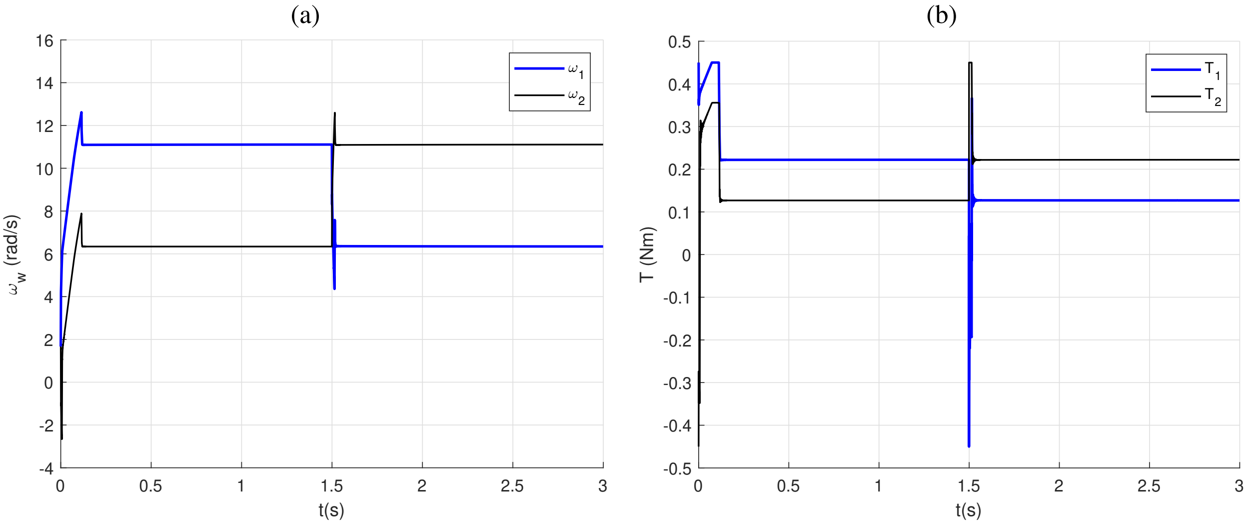


**Figure 8 – circular curves: (a)** $\omega_r$ **(b)** $T$

## CONCLUSIONS

With the rapid advancement of automotive and artificial intelligence technologies, control systems have become prominent in this scenario, especially unmanned and autonomous robotic vehicles. Their applications are numerous, ranging from space exploration, through bomb deactivation and military operations, to simple conveyors in manufacturing lines. Thus, the present work aims to computationally implement and analyze a cascade control strategy based on MPC.

It is noticed that, with the application of this control strategy in cascade, based on MPC and PID, the efficiency in the computational processing is increased, reducing the average processing time of each iteration to approximately one-third in relation to the complete MPC. However, an increase in imprecision is observed in relation to the full MPC, which is not high enough to harm the effectiveness of the control. Thus, the proposed control strategy proved to be very promising for online applications.

For future works, it is suggested to implement the controller in a real rover. It is also suggested to combine the MPC controller with an estimator for tire-ground friction, or another parameter taken as known in this work. In addition, it is possible to study and implement the use of two MPCs in a cascade control, or use the lower PID processing time and give it a smaller control window, updating the data coming from the MPC after a number fixed number of iterations of that.

## REFERENCES


Alatise, M.B. and Hancke, G.P., 2020. “A review on challenges of autonomous mobile robot and sensor fusion methods”. *IEEE Access*, Vol. 8, pp. 39830–39846.

Andersson, J.A., Gillis, J., Horn, G., Rawlings, J.B. and Diehl, M., 2019. “Casadi: a software framework for nonlinear optimization and optimal control”. *Mathematical Programming Computation*, Vol. 11, No. 1, pp. 1–36.

Camacho, E.F. and Alba, C.B., 2013. *Model predictive control*. Springer science & business media.

Cui, Q., Ding, R., Zhou, B. and Wu, X., 2018. “Path-tracking of an autonomous vehicle via model predictive control and nonlinear filtering”. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 232, No. 9, pp. 1237–1252.

Farag, W., 2021. “Complex-track following in real-time using model-based predictive control”. *International Journal of Intelligent Transportation Systems Research*, Vol. 19, No. 1, pp. 112–127.

Negri, G.H., Cavalca, M.S.M. and Celiberto, L., 2017. “Neural nonlinear model-based predictive control with fault tolerance characteristics applied to a robot manipulator”. *Int. J. Innov. Comput. Inf. Control*, Vol. 13, No. 6, pp. 1981–1992.

Ni, J., Hu, J. and Xiang, C., 2021. “A review for design and dynamics control of unmanned ground vehicle”. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 235, No. 4, pp. 1084–1100.

Piedbœuf, J.C., De Carufel, J. and Hurteau, R., 2000. “Friction and stick-slip in robots: simulation and experimentation”. *Multibody System Dynamics*, Vol. 4, No. 4, pp. 341–354.

Škrjanc, I. and Klančar, G., 2017. "A comparison of continuous and discrete tracking-error model-based predictive control for mobile robots". *Robotics and autonomous systems*, Vol. 87, pp. 177–187.

Song, X., Shao, Y. and Qu, Z., 2019. "A vehicle trajectory tracking method with a time-varying model based on the model predictive control". *IEEE Access*, Vol. 8, pp. 16573–16583.

Wang, H., Liu, B., Ping, X. and An, Q., 2019. "Path tracking control for autonomous vehicles based on an improved mpc". *IEEE Access*, Vol. 7, pp. 161064–161073.

Wong, J.Y., 2022. *Theory of ground vehicles*. John Wiley & Sons.

## RESPONSIBILITY NOTICE

The author(s) is (are) the only responsible for the printed material included in this paper.