

# Navigation of a Differential Drive Mobile Robot Using Nonlinear Model Predictive Control

Welid Benchouche<sup>1</sup>, Rabah Mellah<sup>1</sup>, and Mohammed Salah Bennouna<sup>2</sup>

<sup>1</sup>L2CSP Laboratory, Faculty of Electrical and Computing Engineering, University Mouloud Mammri of Tizi-Ouzou, Algeria

<sup>2</sup>Mechanical engineering dept, University Kasdi Merbah, Ouargla, Algeria

welidbenchouche@gmail.com, mellah.rab@gmail.com, bennounams@yahoo.fr

**Abstract**—In this paper, an implementation of a very fast nonlinear model-based predictive controller using a newly developed open-source toolkit (CasADi) was used to attain the two control goals of differential drive mobile robots, point stabilization (regulation) and trajectory following (time-varying reference). The controller's stability was assured by the addition of final state equality constraints, which in general require a long optimization horizon for feasibility. In the work presented here, we performed a full-scale simulation proving the applicability of the terminal stabilization equality constraint have been performed. The obstacle avoidance problem has been solved by adding the obstacle position as a constraint in the main optimal control problem.

**Index Terms**—Nonholonomic robot, Regulation, Trajectory tracking, Navigation, Nonlinear model predictive

## I. INTRODUCTION

The issues of control associated with the class of differential drive mobile robots have attracted the researchers' attention over the last three decennaries. This interest is by cause of theoretical and practical concerns; the nature "non-holonomy" of this category of mobile robot forces limits on velocities acceptable by the system [1]. Notwithstanding, non-holonomy turns out to be valuable because it decreases the control inputs' number, while at the same time, maintaining the system fully controllable in the state space [2]. This benefit, however, introduces a complexity that is related to first control objective: the task cannot be attained with a pure feedback [1]. In the literature, the main control goals considered for this type of systems are point stabilization (regulation) which include forward and parallel parking, and trajectory tracking control. First one aims at driving the robot from one pose to another one. Second one aims to force the robot to follow a provided trajectory changing in time. As it is comparatively simpler, roboticists widely studied the problem of tracking. Researchers in [3] have carried out several tracking techniques, counting feedback linearization, sliding mode, back-stepping and discontinuous methods. In [4], several strategies using Lyapunov control, smooth time-varying control, nonlinear geometric control, piecewise-continuous feedback, and dynamic feedback linearization have been reported. In [2], [5]–[7], Methods fulfilling both control goals were achieved. Among the best cutting-edge control techniques utilized in the industry is model predictive control (MPC); the aim of the latter is to measure a future sequence of control in a given time horizon

in order to guide the prediction of the controlled system's output close to the reference value by means of minimizing an objective function over an online optimization phase in relation to future control actions, as a result, a set of command actions and constraints of the states of the system are met [8]. To resolve the two key control goals of nonholonomic differential drives, variations of MPC (linear MPC and nonlinear (NMPC)) have been utilized. Linear version uses a linearized model of the robot dynamic motion equations to allow it to be used exclusively for the problem of trajectory tracking [9], [10]; or its time-non-parameterized case known as path tracking [8], [11]. Nonlinear version includes the nonlinearities of the system and utilizes the motion model of the robot explicitly in the optimization control problem, in [3], [12], researchers have utilized it for tracking; [4], [13] for regulation; and [14] for both. The usage of a predictive control horizon raises a stability problem as mentioned in [4]. It has been shown that stability can be secured by using final-state equality constraints for a finite receding horizon [15], [16]. Further analysis shows that by adding a final-state penalty, the final-state equality limit can be relaxed as a final-state inequality [3], [4]. Another stability criterion was introduced in [14], centered on first-state contractive predictive control. In [3], [4], and [15], it was stated that it is time-consuming and a practically unmanageable task to achieve stability using the terminal equality constraint. None the less, many dynamic optimization packages that implements nonlinear model predictive control have been developed due to developments in hardware and the development of successful numerical algorithms [17], [18]. In comparison with the formerly developed optimization packages, a lately developed package (CasADi) [19], which easily implements NMPC problems, has been shown to be a free software, user-friendly, extensible, and computationally scalable [19]. It has been noted, as per the outlined literature above, that a study using real-time NMPC is needed for the two key control goals of non-holonomic mobile robots, where stabilizing final equality constraints are considered. Navigation is one of the most important problems in the design and development of intelligent mobile robots, it is the ability of a robot to plan and execute collision-free motions within its environment. The sensor data feeds the controller from the environment. This information is considered in the computation of the control law, in which the vicinity to any

obstacle is also penalized [20]. In other words, the obstacle avoidance problem is solved by considering its position as a constraint in the optimal control problem. The problem that is heightened in this paper is that of driving a differential drive to stabilize at a goal position or follow a formerly calculated desired trajectory, avoiding fixed obstacles: A predictive control strategy would seem to be an acceptable approach to the issue, since we know the desired future reference. In this work, our main contribution is to propose a NMPC that utilizes a final state constraint for stability, a norm 2 distance between the robot pose and the obstacle pose constraint for the obstacle avoidance navigation task. A nonlinear model of the robot kinematics is utilized. control variables constraints are also considered and a quadratic objective function is proposed for computing the set of control signals, and the latter is solved using the multi-shooting technique in CasADi toolbox as it reduces the simulation time by 10 to 20 times compared with single shooting technique the solver used for the OCP is the IPOPT.

## II. MODELLING

In this section, A brief overview of the differential drive kinematics is given along with an example of both control objectives namely regulation and tracking.

### A. Robot Kinematics

The derivations of the WMR's kinematics utilized in this paper are based on the assumptions below [3]: Design Assumptions:

- The WMR does not contain flexible parts; it is considered a rigid body robot
- There is no steering link per wheel; wheels can either go forward or backward only

Operational Assumptions:

- The WMR moves on a planar surface.
- The translational friction at the point of contact between a wheel and the surface is large enough so that no translational slip may occur.

The relevant variables for the kinematic model of a typical two-wheel differential mobile robot are its center position coordinates  $(x_{rob}; y_{rob})$ , its angle of orientation  $\theta_{rob}$ , along with its linear and angular velocities  $(v_{rob}, w_{rob})$ , respectively. To illustrate the position of the robot on the plane, we establish a relationship between the global reference frame on the plane and the local reference frame on the robot. These frames are shown in Fig. 1. Axes  $X_I$  and  $Y_I$  define the global reference frame.  $O$  is the origin. We choose a point  $P$  to represent the position reference point of the robot chassis; the pair  $[X_R, Y_R]^T$  represents the robot reference frame point, also referred to as the local frame point, this pair pass-through the point  $P$  and defines the WMR's local reference frame. Therefore, the classic prediction model derived from Fig. 1 result in the prediction model described in "(1)".

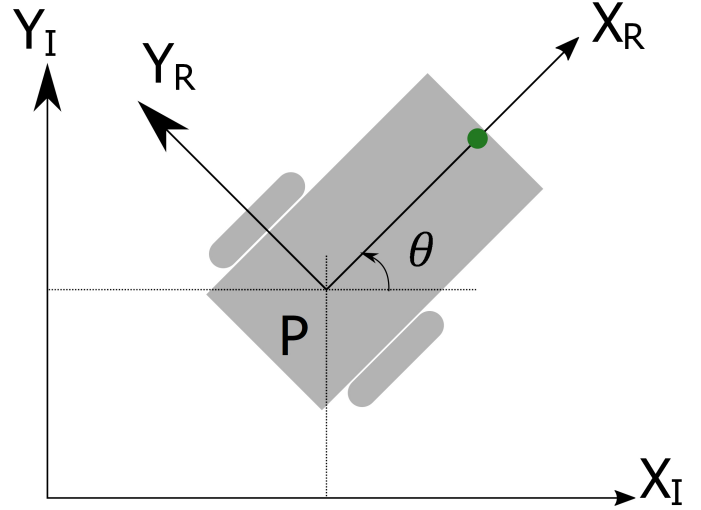


Fig. 1. The global reference frame and the robot local reference frame.

Where, the state and control signal vectors are expressed as  $q = (x_{rob}, y_{rob}, \theta_{rob})^T$  and  $u = (v_{rob}, w_{rob})^T$ .

$$\dot{q} = \begin{bmatrix} \dot{x}_{rob} \\ \dot{y}_{rob} \\ \dot{\theta}_{rob} \end{bmatrix} = \begin{bmatrix} v_{rob} \cos(\Theta) \\ v_{rob} \sin(\Theta) \\ w_{rob} \end{bmatrix} = \begin{bmatrix} \cos(\Theta) & 0 \\ \sin(\Theta) & 0 \\ 0 & 1 \end{bmatrix} u \quad (1)$$

Since "(1)" has the form of a driftless system "(2)" and the condition of the accessibility rank is satisfied globally [21], controllability of "(1)" is assured.

$$\dot{q} = \begin{bmatrix} \dot{x}_{rob} \\ \dot{y}_{rob} \\ \dot{\theta}_{rob} \end{bmatrix} = \begin{bmatrix} \cos(\Theta) \\ \sin(\Theta) \\ 0 \end{bmatrix} v_{rob} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w_{rob} \quad (2)$$

### B. Point Stabilization and Trajectory Tracking

To describe the two control problems at hand, we define a reference robot according similar to "(1)" and subject to the same constraints, where  $q_{ref} = (x_{ref}, y_{ref}, \Theta_{ref})^T$  is the reference state vector, and  $u_{ref} = (v_{ref}, w_{ref})^T$  is the input control vector, thus, the reference robot can be described as:

$$\dot{q}_{ref} = \begin{bmatrix} \dot{x}_{ref} \\ \dot{y}_{ref} \\ \dot{\theta}_{ref} \end{bmatrix} = \begin{bmatrix} v_{ref} \cos(\Theta_{ref}) \\ v_{ref} \sin(\Theta_{ref}) \\ w_{ref} \end{bmatrix} = \begin{bmatrix} \cos(\Theta_{ref}) & 0 \\ \sin(\Theta_{ref}) & 0 \\ 0 & 1 \end{bmatrix} u_{ref} \quad (3)$$

At this level, if the reference vector  $q_{ref}$  has a steady value comparable to the goal position, we are dealing with the point stabilization problem, and the control vector  $u_{ref} = (v_{ref}, w_{ref})^{ref} = (0, 0)^T$ . On the other hand, according to the chosen reference trajectory, if the vectors  $q_{ref}$  and  $u_{ref}$  have values changing in time, we are dealing with the trajectory tracking problem. In both cases, controlling "(1)" to track "(3)" is our goal; hence, we define the tracking error model  $q_e$  in

the basis of the frame linked to the mobile platform or the local frame as follows:

$$q_e = \begin{bmatrix} x_e \\ y_e \\ \Theta_e \end{bmatrix} = \begin{bmatrix} \cos(\Theta) & \sin(\Theta) & 0 \\ -\sin(\Theta) & \cos(\Theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{ref} - x_{rob} \\ y_{ref} - y_{rob} \\ \Theta_{ref} - \Theta_{rob} \end{bmatrix} \quad (4)$$

It can be easily shown that by forcing the state vector  $q_e$  to 0, the two control objectives can be accomplished. We differentiate “(4)” versus time; we get the error dynamics for the tracking problem which is as follows:

$$\begin{aligned} \dot{x}_e &= w_{rob}y_e - v_{rob} + v_{ref}\cos(\Theta_e) \\ \dot{y}_e &= -w_{rob}x_e + v_{ref}\sin(\Theta_e) \\ \dot{\Theta}_e &= w_{ref} - w_{rob} \end{aligned} \quad (5)$$

Linearizing “(5)”, the error model takes the form:

$$\dot{q}_e = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{\Theta}_e \end{bmatrix} = \begin{bmatrix} 0 & w_{ref} & 0 \\ -w_{ref} & 0 & v_{ref} \\ 0 & 0 & 0 \end{bmatrix} q_e + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} u_e \quad (6)$$

where  $u_e$  is:

$$u_e = \begin{bmatrix} -v_{rob} + v_{ref}\cos(\Theta_e) \\ w_{ref} - w_{rob} \end{bmatrix} \quad (7)$$

It can be readily tested that if  $[v_{ref}, w_{ref}] = [0, 0]$ , the controllability of the model “(6)” is lost, eliminating the ability of applying the point stabilization. Now here, it must be stated that in [4], and [14], a motion model version for “(5)” described in the polar coordinates was also used for the purpose of point stabilization control; however, model “(5)” was only used in the present work to attain the two control problems. Thus, no modification to the controller is needed for our implementation.

### III. DESIGN OF MODEL BASED PREDICTIVE CONTROL

Next, an overview of the nonholonomic mobile robots’ scheme of NMPC is provided, outlining the premises that are required for the stability evidence of terminal equality constraints.

#### A. NMPC Controller Design

Generally, it is possible to express “(1)” in a compact form as follows:

$$\dot{q}(t) = f(q(t), u(t)) \quad (8)$$

Where the state is of n-dimension  $q(t) \in R^n$  and the control is of m-dimension  $u(t) \in R^m$ . The purpose of the control is calculating a permissible command  $u(t)$  forcing “(8)” to shift towards the equilibrium defined by ( $q_e(t) = 0$  and  $u_e(t) = 0$ ), where  $q_e = q - q_{ref}$  and  $u_e = u - u_{ref}$ . The purpose of the control process is to minimize the cost function given in [3] as follows:

$$J(t, q_e(\tau), u_e(\tau)) = \int_t^{t+T} l(\tau, q_e(\tau), u_e(\tau)) d\tau \quad (9)$$

Where  $l(\tau, q_e(\tau), u_e(\tau)) = q_e(\tau)^T Q q_e(\tau) + u_e(\tau)^T R u_e(\tau)$  is called the operating cost, and  $T$  is the prediction horizon

and matrices  $Q$  and  $R$  are  $(n \times n)$  and  $(m \times m)$  respectively, they are symmetric positive definite weight matrices. It has been shown that the stability of predictive regulation is ensured by the imposition of terminal equality constraints [15], [16]. Therefore, at time instant  $t$ , the online open-loop optimization problem of our NMPC controller can be illustrated as:

$$\begin{aligned} &\min_u J(t, q_e(\tau), u_e(\tau)) \\ &\text{under the constraints,} \\ &\dot{q}(\tau) = f(x(\tau), u(\tau)) \\ &u(\tau) \in U (\forall \tau \in [t, t+T]) \\ &q_e(t+T) = 0 \end{aligned} \quad (10)$$

Where  $0 \in U \in R^m$  is a contract set defining upper and lower bounds of the control effort, and  $q_e(t+T) = 0$  describes the equality constraints of the terminal state. Controller stability can be proven, as illustrated in [16], if the following two assumptions are fulfilled.

- The state vector  $q_r \in X$  is an equilibrium point where  $q_r$  is the reference state vector for the admissible control value  $u_r \in U$ , and  $X \in R^n$  is the state space set for the state vector  $q(t)$ ; this means that there is a control value  $u_r \in U$  such that  $f(q_r, u_r) = q_r$ .
- The function of running cost  $l : X \times U \rightarrow R_0^+$  fulfils  $l(q_r, u_r) = 0$  from  $u_r \in U$  obtained from the first assumption.

For the first assumption, observing system “(1)” will easily verify these assumptions, and since the operating cost function  $l$  has the quadratic form presented in “(9)”, the second assumption is also fulfilled.

#### B. Obstacle Avoidance

To achieve obstacle avoidance, for the Euclidean distance between the prediction of the location of the robot and the location of the obstacle, we must maintain a lower limit. Thus, we ought to enforce the following path restrictions.

$$\sqrt{(x_{rob} - x_{obs})^2 + (y_{rob} - y_{obs})^2} \geq r_{rob} + r_{obs} \quad (11)$$

This leads to the inequality constraint

$$-\sqrt{(x_{rob} - x_{obs})^2 + (y_{rob} - y_{obs})^2} + (r_{rob} + r_{obs}) \leq 0 \quad (12)$$

The OCP becomes

$$\begin{aligned} &\min_u J(t, q_e(\tau), u_e(\tau)) \\ &\text{under the constraints,} \\ &\dot{q}(\tau) = f(x(\tau), u(\tau)) \\ &u(\tau) \in U (\forall \tau \in [t, t+T]) \\ &q_e(t+T) = 0 \\ &-\sqrt{(x_{rob} - x_{obs})^2 + (y_{rob} - y_{obs})^2} + (r_{rob} + r_{obs}) \leq 0 \end{aligned} \quad (13)$$

In the next section, the code that has been created will initialize the NMPC execution routine each time the simulations are analysed.

#### IV. CASADI TOOLKIT

In this paper, a toolbox (CasADi) of MATLAB was used to simulate our two discussed control objectives, It is an open-source tool for nonlinear optimization and algorithmic differentiation. It facilitates rapid -yet efficient- implementation of different methods for numerical optimal control, both in an offline context and for nonlinear model predictive control (NMPC). In order to be able to meet the final equality constraints set out in the previous section, it is important to maintain prudently the required number of steps of the prediction horizon to satisfy constraints though sustaining a relatively abrupt update speed for controllers, making the online optimization feasible. The used toolkit allows certain stability requirements to be used, as presented in the next section. The NMPC problem that is solved by (CasADi) is of the general form [19]:

$$\begin{aligned} \min_{q(\cdot), u(\cdot), p} & \int_{t_0}^{t_0+T} h(q(t), u(t), p) - \eta(t) \Big|_Q^2 dt \\ & + m(q(t_0 + T), p, t_0 + T) - \mu \Big|_P^2 \\ & \text{under the constraints,} \\ & q(t_0) = q_0 \\ \forall t \in [t_0, t_0 + T] & = f(t, q(t), \dot{q}(t), u(t), p) \\ \forall t \in [t_0, t_0 + T] & \geq s(t, q(t), u(t), p) \\ 0 & = r(q(t_0 + T), p, t_0 + T) \end{aligned} \quad (14)$$

Where  $q(\cdot)$  denotes the states,  $u(\cdot)$  denotes the manipulated variable,  $p$  is a constant parameter of time (optional),  $T$  is the prediction horizon,  $f(\cdot)$  is the system equation,  $s(\cdot)$  the trajectory constraints, and  $r(\cdot)$  is the final state constraints. Furthermore, the cost function is stated in the form of a least squares, where  $\eta$  and  $\mu$  indicate the tracking and final reference. The moving objective function  $\|h(\cdot) - \eta(t)\|_Q^2$  must be stated here exclusively in the work proposed herein. (i.e.,  $\|m(\cdot) - \mu\|_P^2 = 0$ ). It is simple to fit the control problem “(14)” written in scalar notation with the control problem stated in “(13)”.

#### V. SIMULATION RESULTS

This section is devoted for the two control objectives simulation results. Data used here is from the i90 robot from DrRobot company, the latter is a mobile robot platform dedicated to simulation results shown in Fig. 2 and Fig. 3. The robot is a 2 wheeled differential drive mobile robot with an integrated computer, Wi-Fi communications, ultrasonic scanners, IRs, camera, and additional autonomous functions. The robot utilizes a high precision 40 kg.cm 12V DC motor with embedded 800 Count per Cycle Optical Encoder tick encoders for localization purposes. A 2.5 GHz i5 6300 computer with a 8 GB of memory, Windows 10 operating system was utilized to perform the control vector  $u$  computation. Full robot specifications can be found in [22].

The online optimisation parameters are selected to achieve the satisfactory controller efficiency including the sampling frequency, the number of prediction horizon steps ( $N$ ), and the weight matrices  $Q$  and  $R$  are given in “(9)”. In the next, the



Fig. 2. The i90 robot from DrRobot company

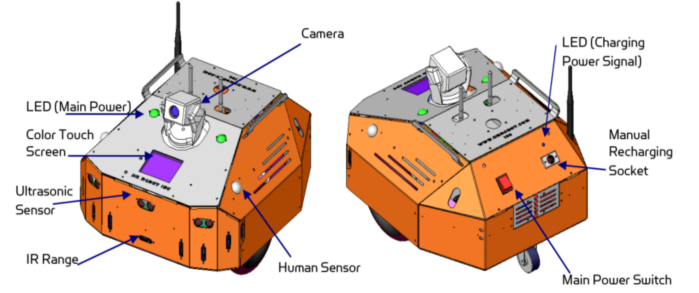


Fig. 3. The i90 robot specifications

results of the stabilization point are discussed first, followed by the results of the trajectory.

##### A. Point stabilization navigation results

The efficiency of the NMPC controller for point stabilization is seen in this section. The robot begins with the initial pose  $q_0 = [000]^T (m, m, rad)$ . The robot is commanded to stabilize at the pose  $x_r = [8.59.50]^T$ . The controller sampling time step is 0.2seconds and the number of prediction steps number is  $N = 20$ , resulting in a prediction horizon time  $T = 10seconds$ . The weighting matrices  $Q$  and  $R$  of the objective function “(9)” are diagonal matrices with diagonal elements (15, 10, 0.1) for  $Q$ , and (0.002, 0.002) for  $R$ . The controller saturation limits for linear velocity  $v$  and angular velocity  $w$  commands are set to ensure accurate location of the robot and to meet its actuators saturation limits as follows [23]:

$$\begin{bmatrix} -0.75 \\ -753/767 \end{bmatrix} \leq \begin{bmatrix} v(ms) \\ w(rads) \end{bmatrix} \leq \begin{bmatrix} 0.75 \\ 753/767 \end{bmatrix} \quad (15)$$

A summary of the results is in Fig. 4 and Fig. 5. Fig. 4 demonstrates the trajectories performed by the robot (in red).

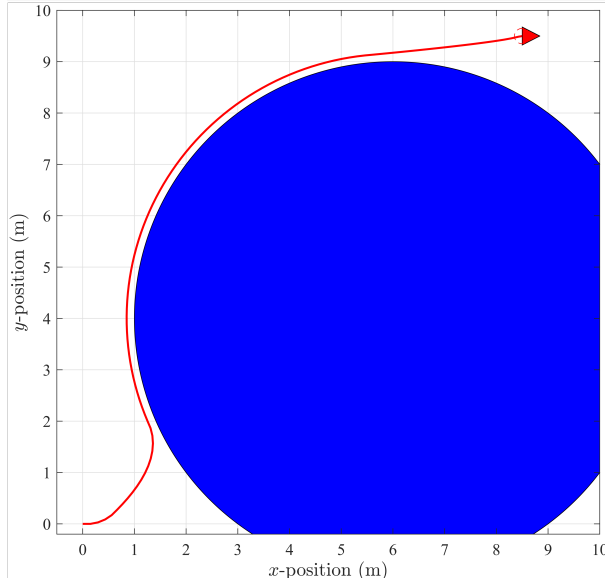
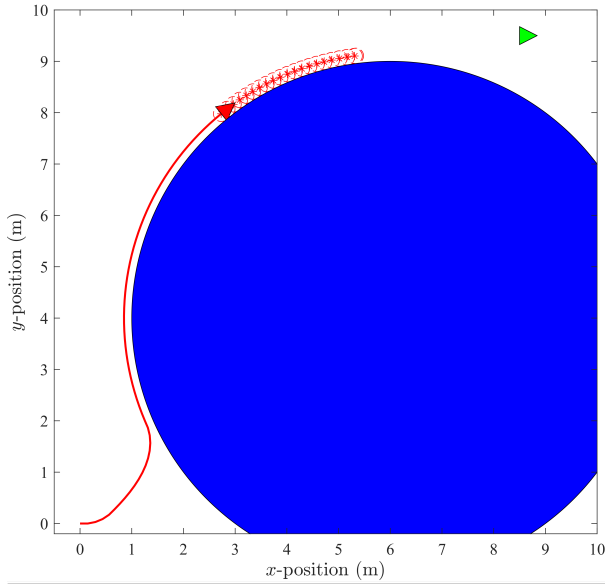


Fig. 4. Point stabilization trajectory results

The red triangle shows the position and orientation of the robot, the circles show the prediction of the state, the obstacle is represented by a blue disk and the reference position by a green triangle. As shown in the Fig. 4, the robot can stabilize the controller perfectly to the required position. The results demonstrate the controller's satisfactory performance in this case. Fig. 5 demonstrates that the robot's actual linear and angular speeds meet the saturation limit specified in "(15)".

### B. Trajectory tracking navigation results

For the trajectory tracking problem, the performance of the nonlinear model predictive controller was illustrated by the consideration of two reference paths, a circular and an 8 shape path, namely "(16)", "(17)" and "(18)", "(19)". Parameters of the "(16)", "(17)" and "(18)", "(19)" trajectories are selected so that  $v_r$  and  $w_r$  do not violate "(15)", for the circular path,

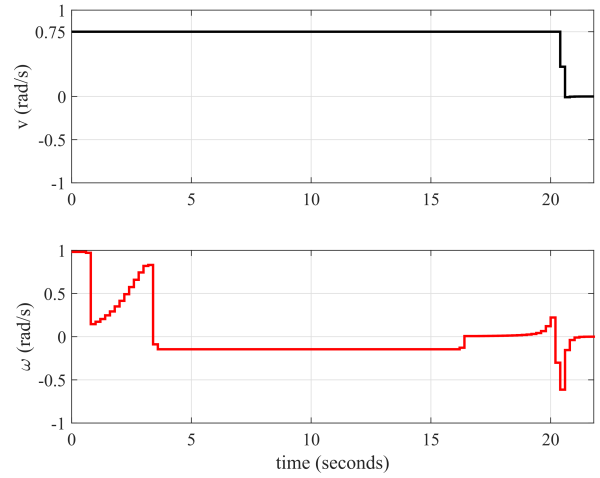


Fig. 5. Point stabilization robot velocities

the initial pose of the robot is  $q_0 = [0, 0, 0]^T(m, m, rad)$ , and for the eight-shaped path the initial pose is  $q_0 = [1, 2, 0]^T(m, m, rad)$

$$x_{ref}(t) = 0.3 + 2\sin(0.25t) \quad (16)$$

$$y_{ref}(t) = -2.3 + 2\cos(0.25t) \quad (17)$$

$$x_{ref}(t) = 0.3 + 1.5\sin(0.3t) \quad (18)$$

$$s_{ref}(t) = 0.3 + 2.5\cos(0.15t) \quad (19)$$

The controller sampling time step is  $0.2seconds$  with  $N = 20$  the prediction steps, resulting in a prediction horizon time  $T = 4seconds$ . The starting points of the reference trajectories "(16)", "(17)" and "(18)", "(19)" have been selected so that an initial error is different than 0. For the circular-shape tracking, the weight matrices  $Q$  and  $R$  of the objective function "(9)" are diagonal matrices with diagonal elements defined as  $(30, 30, 0.2)$  for  $Q$ , and as  $(50, 50)$  for  $R$ .

For the eight-shape tracking, the diagonal elements of  $Q$  are  $(30, 30, 0.2)$ , and the diagonal elements of  $R$  are  $(9, 5)$ . Fig. 6 and Fig. 8 show the actual robot trajectory, the robot trajectory is presented in red, the reference trajectory in dashed blue, the predicted state is shown in red circles and the obstacles in blue disks. Fig. 7 and Fig. 9 show the linear and angular control signals, resulting from following the reference trajectory.

It can be seen from the four figures Fig. 6, Fig. 7, Fig. 8, and Fig. 9 that the errors in the state vector  $q_e$  are maintained within acceptable limits, whereas the robot speeds are maintained within the limits given in "(15)".

The average computational cost per time step for circular-shape tracking is  $(24.2ms)$ , the steady-state value of the position error was observed to be within  $(\pm 3.5cm)$  and the orientation error was observed to be within  $(\pm 0.04rad)$ , except when the robot avoids obstacles.

In the 8-shape tracking, the average time computational costs  $(23.2ms)$  have been shown to be within  $(\pm 4.7cm)$  and orientation errors  $(\pm 0.065rad)$ , except when the robot is avoiding obstacles.

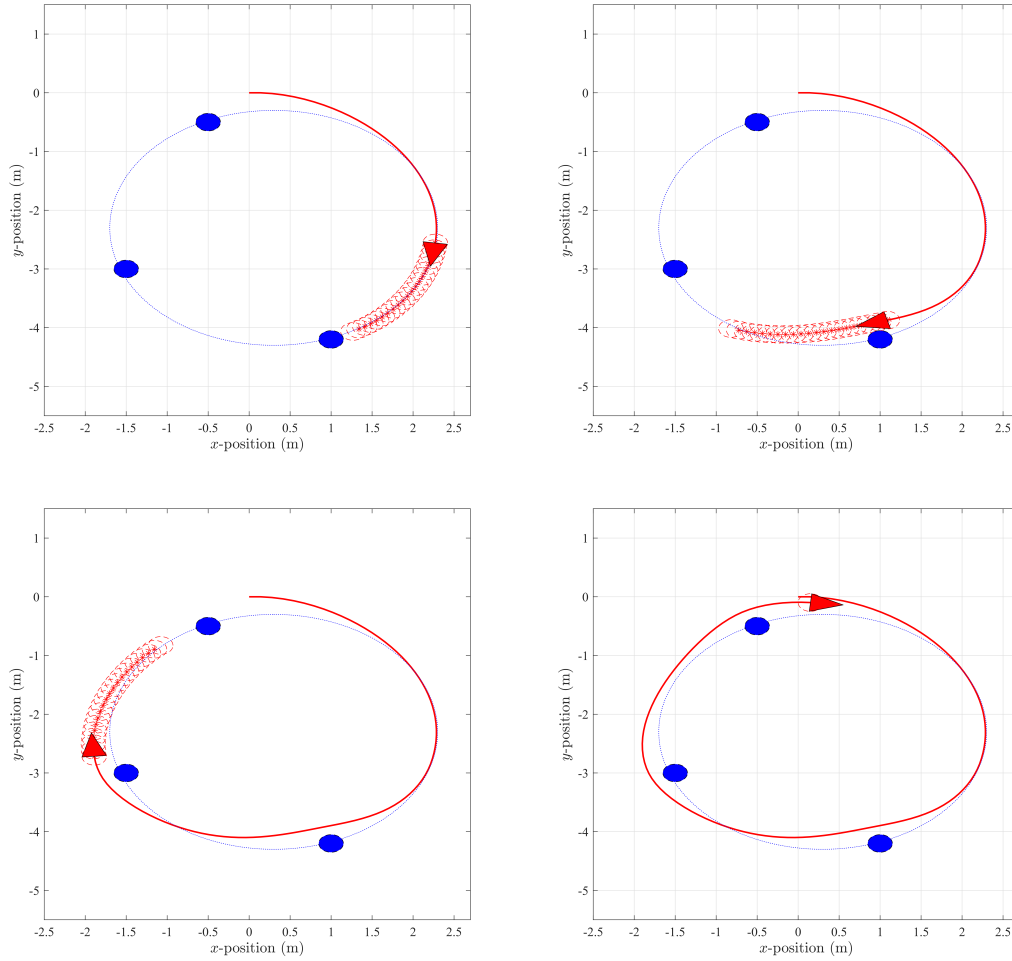


Fig. 6. Circular shape trajectory tracking and navigation results

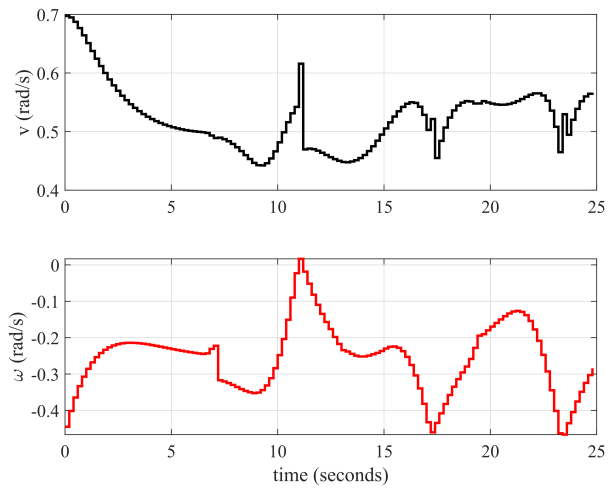


Fig. 7. Circular shape trajectory tracking and navigation velocities results

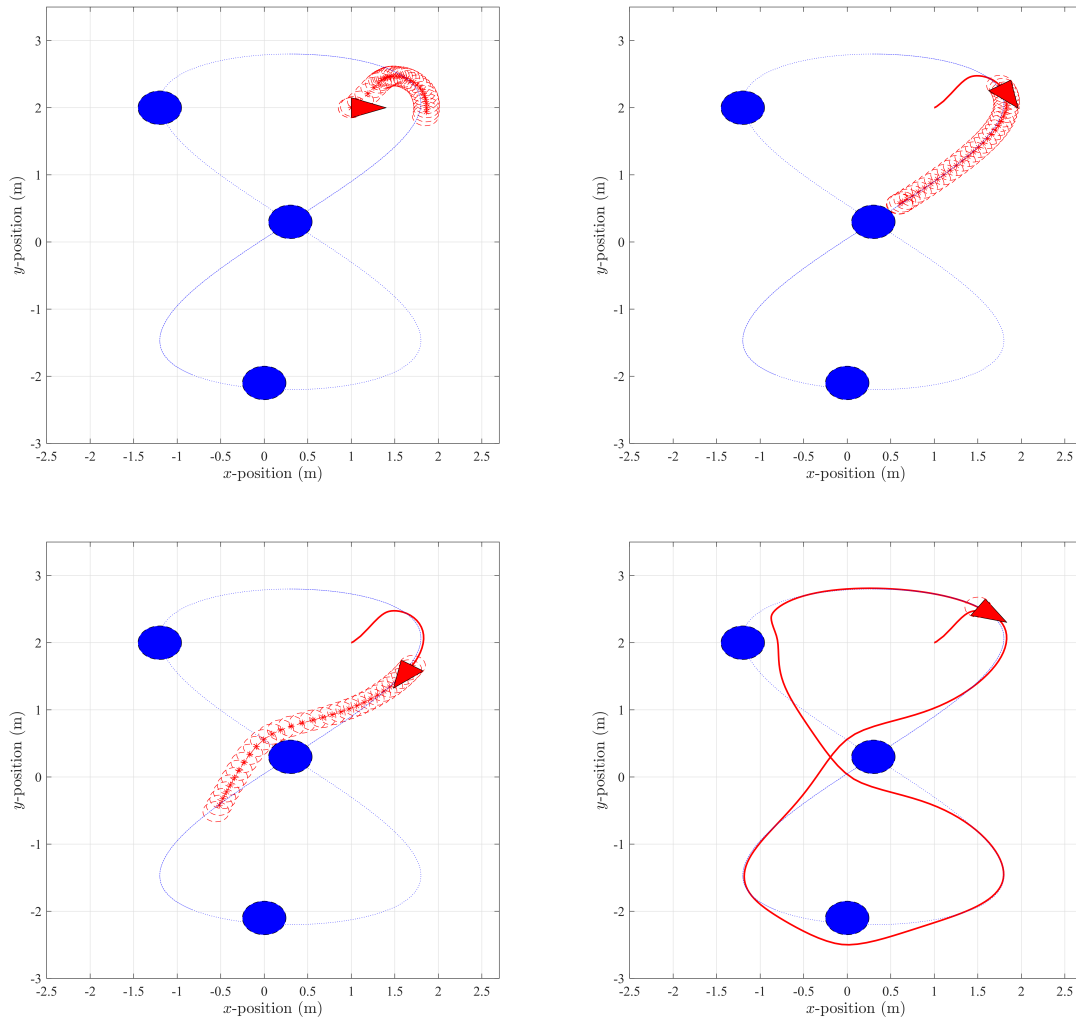


Fig. 8. Lemniscates shape trajectory tracking and navigation results

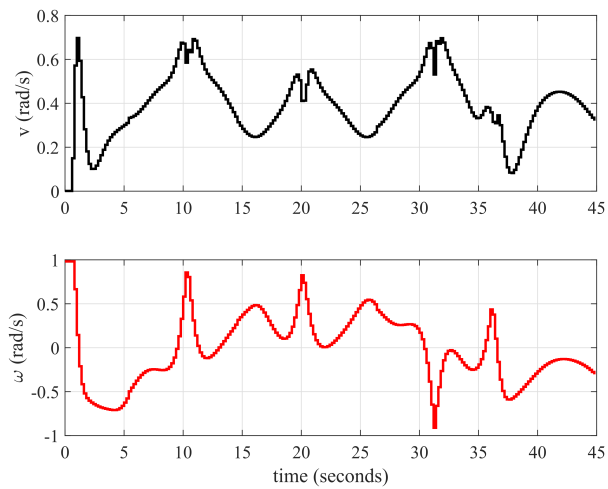


Fig. 9. Lemniscates trajectory tracking and navigation velocities results

## VI. CONCLUSION

In this paper, a stabilizing NMPC controller with a terminal equality constraints has been shown to be applicable, for 2 control objectives namely, point stabilization and trajectory tracking of mobile robots. Usage of a toolkit that implements fast NMPC routines, making tractable the computationally challenging stability requirements. Simulations were conducted using I90 DrRobot research platform specifications. In the point stabilization case, the robot was commanded to perform parallel parking, whereas in the trajectory tracking problem, the robot was commanded to follow circular and eight-shape trajectories. The obtained results showed a satisfactory controller performance in terms of the point stabilization and tracking errors, hence, proving the applicability of the stabilizing NMPC scheme.

## REFERENCES

- [1] J.-P. Laumond, S. Sekhavat, and F. Lamiroux, "Guidelines in nonholonomic motion planning for mobile robots," *In Robot motion planning and control*, pp. 1–53, 1998.
- [2] M. Michalek and K. Kozłowski, "Vector-field-orientation feedback control method for a differentially driven vehicle," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 1, pp. 45–65, 2009.
- [3] D. Gu and H. Hu, "Receding horizon tracking control of wheeled mobile robots," *IEEE Transactions on control systems technology*, vol. 14, no. 4, pp. 743–749, 2006.
- [4] D. Gu and H. Hu, "A stabilizing receding horizon regulator for nonholonomic mobile robots," *IEEE Transactions on Robotics*, vol. 21, no. 5, pp. 1022–1028, 2005.
- [5] W. E. Dixon, D. M. Dawson, F. Zhang, and E. Zergeroglu, "Global exponential tracking control of a mobile robot system via a pe condition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 30, no. 1, pp. 129–142, 2000.
- [6] G. Oriolo, A. De Luca, and M. Vendittelli, "Wmr control via dynamic feedback linearization: design, implementation, and experimental validation," *IEEE Transactions on control systems technology*, vol. 10, no. 6, pp. 835–852, 2002.
- [7] T.-C. Lee, K.-T. Song, C.-H. Lee, and C.-C. Teng, "Tracking control of unicycle-modeled mobile robots using a saturation feedback controller," *IEEE transactions on control systems technology*, vol. 9, no. 2, pp. 305–318, 2001.
- [8] G. V. Raffo, G. K. Gomes, J. E. Normey-Rico, C. R. Kelber, and L. B. Becker, "A predictive controller for autonomous vehicle path tracking," *IEEE transactions on intelligent transportation systems*, vol. 10, no. 1, pp. 92–102, 2009.
- [9] P. Falcone, M. Tufò, F. Borrelli, J. Asgari, and H. E. Tseng, "A linear time varying model predictive control approach to the integrated vehicle dynamics control problem in autonomous systems," *In 2007 46th IEEE Conference on Decision and Control*, pp. 2980–2985, 2007.
- [10] G. Klančar and I. Škrjanc, "Tracking-error model-based predictive control for mobile robots in real time," *Robotics and autonomous systems*, vol. 55, no. 6, pp. 460–469, 2007.
- [11] J. Backman, T. Oksanen, and A. Visala, "Navigation system for agricultural machines: Nonlinear model predictive path tracking," *Computers and Electronics in Agriculture*, vol. 82, pp. 32–43, 2012.
- [12] H. Lim, Y. Kang, C. Kim, J. Kim, and B.-J. You, "Nonlinear model predictive controller design with obstacle avoidance for a mobile robot," *In 2008 IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications*, pp. 494–499, 2008.
- [13] F. Kuhne, W. F. Lages, and J. G. Da Silva, "Point stabilization of mobile robots with nonlinear model predictive control," *In IEEE International Conference Mechatronics and Automation, 2005*, vol. 3, pp. 1163–1168, 2005.
- [14] F. Xie and R. Fierro, "First-state contractive model predictive control of nonholonomic mobile robots," *In 2008 American Control Conference*, pp. 3494–3499, 2008.
- [15] J. B. Rawlings and K. R. Muske, "The stability of constrained receding horizon control," *IEEE transactions on automatic control*, vol. 38, no. 10, pp. 1512–1516, 1993.
- [16] G. Lars and P. Jürgen, "Nonlinear model predictive control theory and algorithms," 2011.
- [17] D. B. Leineweber, I. Bauer, H. G. Bock, and J. P. Schlöder, "An efficient multiple shooting based reduced sqp strategy for large-scale dynamic process optimization. part 1: theoretical aspects," *Computers & Chemical Engineering*, vol. 27, no. 2, pp. 157–166, 2003.
- [18] L. L. Simon, Z. K. Nagy, and K. Hungerbuehler, "Swelling constrained control of an industrial batch reactor using a dedicated nmpc environment: Optcon," *In Nonlinear model predictive control*, pp. 531–539, 2009.
- [19] B. Houska, H. J. Ferreau, and M. Diehl, "Acado toolkit—an open-source framework for automatic control and dynamic optimization," *Optimal Control Applications and Methods*, vol. 32, no. 3, pp. 298–312, 2011.
- [20] D. Feng and B. H. Krogh, "Dynamic steering control of conventionally steered mobile robots," *Journal of robotic systems*, vol. 8, no. 5, pp. 699–721, 1991.
- [21] A. Bloch and B. Brogliato, "Nonholonomic mechanics and control," *Appl. Mech. Rev.*, vol. 57, no. 1, pp. B3–B3, 2004.
- [22] "I90drrobot," [http://drrobot.com/products\\_item.asp?itemNumber=i90](http://drrobot.com/products_item.asp?itemNumber=i90), accessed: 2021-02-30.
- [23] M. W. Mehrez, G. K. Mann, and R. G. Gosine, "Stabilizing nmpc of wheeled mobile robots using open-source real-time software," *In 2013 16th International Conference on Advanced Robotics (ICAR)*, pp. 1–6, 2013.