Wheel Slip Avoidance through a Nonlinear Model Predictive Control for Object Pushing with a Mobile Robot

Filippo Bertoncelli * **Fabio Ruggiero** ** **Lorenzo Sabattini** *

* *Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Italy (e-mail: {filippo.bertoncelli, lorenzo.sabattini}@unimore.it).*
** *CREATE Consortium and Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, 80125, Naples, Italy (e-mail: fabio.ruggiero@unina.it).*

Abstract: Wheel slip may cause a significative worsening of control performance during the movement of a mobile robot. A method to avoid wheel slip is proposed in this paper through a nonlinear model predictive control. The constraints included within the optimization problem limit the force exchanged between each wheel and the ground. The approach is validated in a dynamic simulation environment through a Pioneer 3-DX wheeled mobile robot performing a pushing manipulation of a box.

Keywords: Mobile robots, predictive control, manipulation tasks, optimization problems, friction.

1. INTRODUCTION

In this paper, we consider the problem of mobile robots moving an object in the environment. For this purpose, we propose a control strategy based on the nonlinear model predictive control (NMPC) technique. The scenario considered in this paper is that of a large and/or heavy object to be moved in a two-dimensional environment by a mobile robot that performs a pushing manipulation. In the presence of heavy objects, large forces are applied to the pushing mobile robot: this may cause wheel slip, which reduces the precision of the executed motion, and thus the efficiency of the pushing operation. In this paper, we introduce a methodology to prevent wheel slip.

In the literature, manipulation of an object through one (or more) mobile robots is typically addressed with a prehensile approach. This is generally achieved equipping robots with manipulation tools (e.g., grippers), as in [Wang and Schwager, 2016], or achieving force closure using multiple robots displaced around the object to be manipulated, as in [Yamashita et al., 2003]. These approaches resemble the most common solution exploited in robotics for solving the problem of moving an object: the pick-and-place method, where the object is grasped in a stiff manner and is then moved to the desired location. While this is a common and effective solution in several cases, it can not always be applied, in particular when the size of the object is too large, when its shape is unknown a priori, when it is excessively heavy, or when its surface can be damaged by a stiff grasp, as discussed by Lynch and Mason [1996]. Nonprehensile manipulation approaches can then be exploited in these cases as proposed by Ruggiero et al. [2018]. Specifically, these strategies include methods in which the robot imposes the motion to the object through unilateral constraints only, such as in the case of pushing. While nonprehensile manipulation can represent a solution in the scenarios above, its successful implementation requires to take into account the dynamic models of the robot, the object, and the environment, since the exchanged forces are of paramount importance.

Nonprehensile manipulation has been recently implemented equipping the robots with flexible elements, such as ropes or cables. For instance, Kim and Shell [2017] consider a robot equipped with a *tail* (i.e., a flexible cable), and propose a planning method to define the motion of the robot to exploit such a tail for moving an object. While this method has proven its feasibility for simple objects to be manipulated, alternative methods are necessary for more complex scenarios. Objects with general shape can be considered by the method proposed by Maneewarn and Detudom [2005], where two mobile robots are connected employing a cable, for cooperatively pulling a heavy object. The main drawback of this solution is represented by the physical interconnection between the robots (i.e., the cable), that significantly reduces the freedom of motion.

To avoid these issues, mobile robots can be controlled for directly pushing the object to be manipulated, as discussed by Kolhe et al. [2010]. To achieve this objective, it is necessary to guarantee that the mobile robot can move in the environment, without colliding with obstacles, to change its relative position with respect to the object (and, consequently, the pushing direction). This is achieved by Krivic et al. [2016], where uncertainties

⋆ The research leading to these results has been partially supported by the RoDyMan project, which has received funding from the European Research Council FP7 Ideas under Advanced Grant agreement number 320992.

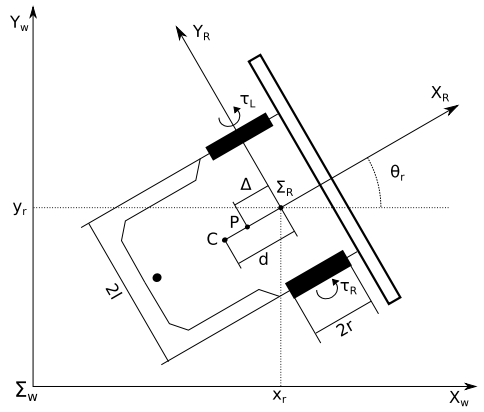


Fig. 1. Schematic representation of the differential drive mobile robot. The black rectangles are the wheels. The black circle is the caster wheel. The frontal bumper is represented by the white rectangle in front of the wheels.

in both the control and the motion execution are dealt with by means of an appropriate motion planning strategy, that considers an increased size of the pushed object, to accommodate re-positioning maneuvers of the pushing robot. A reinforcement learning framework is proposed by Kovac et al. [2004], to define the motion pattern for two robots pushing a box, in a very simplistic scenario, in which dynamics are entirely neglected. A similar case is considered by Igarashi et al. [2010], where an artificial potential field is defined to let a robot, or a group of robots, push an object by just measuring its instantaneous direction of motion. Also in this case, dynamic effects (e.g., friction) are not considered, which makes the proposed method not suitable for complex situations, such as in the presence of non-uniform friction, or when more than two robots are needed. A similar problem is considered by Golkar et al. [2009], where a fuzzy controller is proposed for controlling two robots to push an object with known geometrical properties.

The contribution of this paper is in the definition of a novel formulation, based on the NMPC, about the pushing problem performed by a mobile robot. The proposed formulation allows us to explicitly consider the dynamics of the robot, the object, and the friction. In particular, in this paper, we formulate the constraints that guarantee that wheel slip becomes negligible.

2. MODELING

The dynamic model of the robot is introduced in this section. We consider a differential drive robot moving on a plane, whose schematic representation is depicted in Fig. 1. The robot is equipped with a frontal bumper rigidly attached to its body. Let Σ_w be the fixed world frame, and Σ_r be the body frame attached to the midpoint of the axle of the mobile robot. In addition, let $p_r = [x_r \ y_r]^T \in \mathbb{R}^2$ be the position of Σ_r in Σ_w, $\upsilon \in \mathbb{R}$ and $\omega \in \mathbb{R}$ be the heading and angular velocities of the mobile robot, respectively. Finally, let $\theta_r \in \mathbb{R}$ be the angle expressing the rotation of Σ_r in Σ_w. The kinematic model of the mobile robot can be expressed by the following equations

$$\begin{bmatrix} \dot{x}_r \\ \dot{y}_r \\ \dot{\theta}_r \end{bmatrix} = \begin{bmatrix} \cos\theta_r & 0 \\ \sin\theta_r & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \upsilon \\ \omega \end{bmatrix}. \tag{1}$$

Through standard computations, the dynamic model of the mobile robot was developed by Dhaouadi and Hatab [2013], and it can be expressed as

$$\begin{aligned} \dot{x}_r &= \cos\theta_r v \\ \dot{y}_r &= \sin\theta_r v \\ \dot{\theta}_r &= \omega \\ \Big(m_r + \frac{2I_w}{r^2}\Big)\dot{v}_r &= \frac{1}{r}(\tau_R + \tau_L) - m_c d\omega^2 \\ \Big(I_r + \frac{2l^2}{r^2}I_w\Big)\dot{\omega}_r &= \frac{l}{r}(\tau_R - \tau_L) + m_c \omega v d \end{aligned} \tag{2}$$

with $q = [x_r \ y_r \ \theta_r \ \upsilon \ \omega]^T \in \mathbb{R}^5$ being the state vector of the robot, $m_r \in \mathbb{R}^+$ the total mass of the vehicle, $I_r \in \mathbb{R}^+$ the inertia moment of the robot around the vertical axis, $I_w \in \mathbb{R}^+$ the inertia moment of a wheel about its axis, $l \in \mathbb{R}^+$ the half the wheel separation (see Fig. 1), $r \in \mathbb{R}^+$ the wheel radius, $d \in \mathbb{R}^+$ the distance of the center of mass of the body of the robot on the robot axis (see Fig. 1), and $m_c \in \mathbb{R}^+$ the mass of the body of the robot (i.e., excluding the wheels). The control input is the pair $u = [\tau_R \ \tau_L]^T \in \mathbb{R}^2$, which are the torques acting on the wheels.

3. CONTROLLER DESIGN

In this section, we describe the design of the controller used to command the robot. We choose the NMPC scheme because it allows to include dynamic constraints in the design of the controller explicitly. The main idea of the scheme is to optimize the predicted future behavior of the system over a finite time horizon.

3.1 Problem Formulation

The idea behind the NMPC is the repetitive solution of an optimal nonlinear control problem (NLP). Given the measured state q_0 at each controller time step T_s, the discretized version of the dynamic model is employed by the NMPC to predict the future behavior of the system state $\hat{q}(k)$, with $k = 0, \ldots, N-1$, where $N \geq 2$ is the prediction horizon, and $\hat{q}(0) = q_0$. Such a prediction is useful to optimize the control sequence $u(0), \ldots, u(M-1)$, with $0 < M \leq N$ the control horizon, and $u(i) = u(M-1)$ for $i = M, \ldots, N-1$. The peculiarity of the NMPC algorithm is that only the first element $u(0)$ of the sequence is applied to the real system. The NLP is repeatedly solved from each new acquired measure. The NLP minimizes an objective function, generally composed of the states and the inputs, with respect to the input variable and subject to a set of proper constraints. Nevertheless, if formulated in this way, the NLP becomes of high dimension, and the computation time and the accuracy of the solver worsen. Instead, we use the recursive elimination methodology proposed by Grüne and Pannek [2017]. Such a methodology decouples the dynamic model of the system from the NLP by reducing the size of the optimization variable and allowing each problem to be treated by specialized solution methods. With a slight abuse of notation regarding the dependencies

for each function, the optimization control problem can be written as

$$\begin{aligned} &\text{minimize } J(z) \\ &\text{w.r.t. } z = (u(0)^T, \dots, u(M-1)^T)^T \in \mathbb{R}^{2M} \\ &\text{s.t } G(z) \leq \bar{0}, \end{aligned} \tag{3}$$

where

$$\begin{aligned} J(z) = e(N)^T P e(N) + u(N)^T W_N u(N) \\ + \sum_{k=1}^{N-1} e(k)^T Q e(k) + u(k)^T W u(k) \end{aligned} \tag{4}$$

is the functional cost to minimize, while $Q \in \mathbb{R}^{5\times5}$, $P \in \mathbb{R}^{5\times5}$, $W \in \mathbb{R}^{2\times2}$ and $W_N \in \mathbb{R}^{2\times2}$ are diagonal and positive semi-definite matrices [1]. The error $e(k) \in \mathbb{R}^5$ is intended as the difference between the predicted $\hat{q}(k)$ and the desired state $q_{ref}(k) \in \mathbb{R}^5$. Such an error is expressed in Σ_r and it is calculated as

$$e(k) = R(\theta(k)) \left(\hat{q}(k) - q_{ref}(k)\right), \tag{5}$$

with

$$R(\theta(k)) = \begin{bmatrix} \cos(\theta(k)) & \sin(\theta(k)) & 0 & 0 & 0 \\ -\sin(\theta(k)) & \cos(\theta(k)) & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \in \mathbb{R}^{5\times5}. \tag{6}$$

By denoting with

$$\hat{q}(k+1) = F(q(k), u(k)), \quad k = 0, \dots, N-1, \tag{7}$$

the Euler-discretized version of (2), the robot state prediction is calculated through (7) from the current measure q_0. Finally, the inequality constraints are expressed as $G(z) \leq \bar{0}$, with $\bar{0} \in \mathbb{R}^{13N}$ the zero vector of proper dimension, and $G(z) \in \mathbb{R}^{13N}$ defined as detailed in the next subsection.

3.2 Wheel Slip Constraints

The first subset of constraints is the following

$$\begin{aligned} x_r(k) + x_{lim} \leq 0, \quad &(8a) \qquad \Omega_r(k) + \Omega_{lim} \leq 0, \quad (8e) \\ x_{lim} - x_r(k) \leq 0, \quad &(8b) \qquad \Omega_{lim} - \Omega_r(k) \leq 0, \quad (8f) \\ y_r(k) + y_{lim} \leq 0, \quad &(8c) \qquad \Omega_l(k) + \Omega_{lim} \leq 0, \quad (8g) \\ y_{lim} - y_r(k) \leq 0, \quad &(8d) \qquad \Omega_{lim} - \Omega_l(k) \leq 0, \quad (8h) \end{aligned}$$

for $k = 0, \dots, N-1$. The inequalities (8a)—(8d) limit the position of the robot inside a rectangle of the plane delimited by the coordinates $x_{lim} \in \mathbb{R}$ and $y_{lim} \in \mathbb{R}$ in Σ_w. The inqualities (8e)—(8h) impose instead that the velocities of the wheels remain inside the saturation limit $\Omega_{lim} \in \mathbb{R}$ of the motors.

Beyond the constraints introduced above, the main focus of our work is the design of an explicit condition to avoid wheel slip. Wheel slip is defined as the relative motion between a tire and the surface on which it is moving. This relative motion occurs when the force needed to maintain the contact exceeds the maximum friction force. The Coulomb friction model provides the following

[1] The matrices W and W_N can be zero matrices since we force the control input to be bounded to include the saturation of the actuators explicitly.

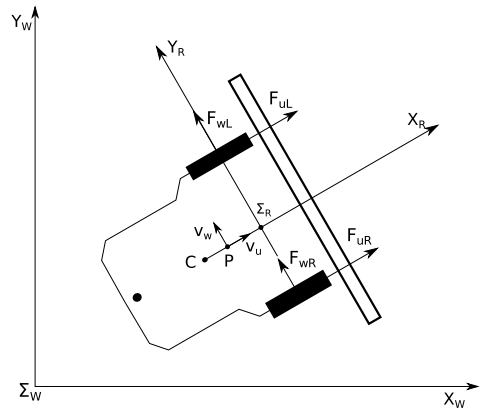


Fig. 2. Planar forces and torques acting on the robot.

representation of the wrench transmitted to an object through a point contact with friction [Murray et al., 1994]

$$F_{c_i} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}^T f_{c_i} \in \mathbb{R}^6, \tag{9}$$

where $f_{c_i} \in FC_{c_i}$ and

$$FC_{c_i} \triangleq \{f = [f_{t1} \;\; f_{t2} \;\; f_n]^T \in \mathbb{R}^3 : \sqrt{f_{t1}^2 + f_{t2}^2} \leq \mu f_n, f_n \geq 0\} \tag{10}$$

is the so-called friction cone, while $f_{t1} \in \mathbb{R}$ and $f_{t2} \in \mathbb{R}$ are the components of a force f tangential to the contact, and $f_n \in \mathbb{R}$ is the component of f normal to the contact with positive sign going inside the object. The friction coefficient between each wheel and the ground is denoted by $\mu \in \mathbb{R}^+$.

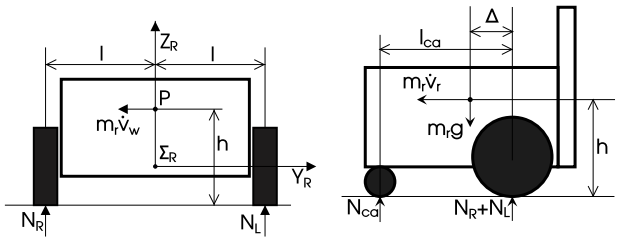
Following such representation, to obtain wheel slip avoidance, we can conclude that the frictional forces need to balance the forces exerted by the robot to achieve a specific motion at any time. The model of the addressed robot interacts with the floor in three points: the two driving wheels and the caster. We assume the planar forces transmitted by the caster wheel are negligible, and we consider the vertical component only. Figure 2 shows the forces exerted by the driving wheels decomposed into the longitudinal (subscript u) and lateral (subscript w) components. To maintain a rolling contact between the wheels and the ground (i.e., to avoid slippage) the following relations need to be verified:

$$\sqrt{F_{uL}^2 + F_{wL}^2} \leq \mu N_L, \quad \sqrt{F_{uR}^2 + F_{wR}^2} \leq \mu N_R. \tag{11}$$

These forces are related to the robot's movement through the following equations:

$$m_r \dot{v}_u = m_r \dot{v}_r = F_{uL} + F_{uR}, \tag{12}$$
$$m_r \dot{v}_w = m_r \dot{\omega}_r \Delta = F_{wR} + F_{wL}, \tag{13}$$
$$I_r \ddot{\theta}_r = I_r \dot{\omega}_r = (F_{uR} - F_{uL}) l + (F_{wL} + F_{wR}) \Delta. \tag{14}$$
$$N_L + N_R + N_{ca} = m_r g, \tag{15}$$
$$N_{ca} l_{ca} - m_r g \Delta + m_r \dot{v}_r (h - r) = 0, \tag{16}$$
$$N_L - N_R = \frac{m_r \dot{v}_w (h - r)}{l}. \tag{17}$$

where (12) and (13) describe the movement along the robot's X_R and Y_R axes respectively and (14) describes the rotation about the Z_R axis. Equation (15) relates the reaction forces of the ground caused by the weight of the



(a) Frontal view of the robot. (b) Lateral view of the robot.

Fig. 3. Forces exchanged between the robot and the ground.

robot. Considering the equilibrium to the rotation about the robot's X_R and Y_R axis, equations (16) and (17) can be derived, these relate forces F_{uR}, F_{uL}, F_{wL}, and F_{wR} with the vertical forces acting on the robot. In particular, Fig (3a) depicts the considered forces in a frontal section of the robot, where $N_L \in \mathbb{R}$ and $N_R \in \mathbb{R}$ are the left and right reaction forces of the wheels with the ground, and $m_r\dot{v}_w$ is the apparent inertial force caused by the robot lateral motion. Conversely, Fig. (3b) depicts the forces in a lateral view of the robot, where $N_{ca} \in \mathbb{R}$ is the reaction force of the caster with the ground, $m_r g$ represents the weight of the robot and $m_r\dot{v}_r$ is the apparent inertial force caused by the longitudinal motion of the robot. Notice that in equations (12)—(17) the forces F_{wL} and F_{wR} appear always in the form $F_{wL} + F_{wR}$ thus it's not possible to extract the different values for the two forces from the system of equations, preventing the direct computation of (11).

If we assume $F_{wL}F_{wR} \geq 0$, the following relations are always verified

$$\sqrt{F_{uL}^2 + F_{wL}^2} \leq \sqrt{F_{uL}^2 + (F_{wL} + F_{wR})^2}, \qquad (18)$$

$$\sqrt{F_{uR}^2 + F_{wR}^2} \leq \sqrt{F_{uR}^2 + (F_{wL} + F_{wR})^2}. \qquad (19)$$

Such an assumption implies that there is no relative motion between the wheels on the Y_RZ_R-plane of the robot. The truthfulness of (18)—(19) allows us to write the following two constraints which imply (11) by the transitive property of inequalities.

$$\sqrt{F_{uL}^2 + (F_{wL} + F_{wR})^2} \leq \mu N_L, \qquad (20)$$

$$\sqrt{F_{uR}^2 + (F_{wL} + F_{wR})^2} \leq \mu N_R. \qquad (21)$$

Since it is possible to extract N_L,N_R,N_{ca},F_{uR},F_{uL} and F_{wR}+F_{wL} from (12)—(17) the constraints are computable from the controller point of view. It is worth noting that the solution extracted from the system of equations above can assume non-physical values unless we apply the following constraints to the problem

$$N_L \geq 0, \qquad N_R \geq 0, \qquad N_{ca} \geq 0. \qquad (22)$$

These relations also imply that the wheels remain on the ground at all times. The relations (20)—(22) are added to the inequality constraints of the problem (3). The final inequality constraint vector $G(z)$ is composed by (8a)—(8h) and (20)—(22), and it is reiterated for each time step of the prediction horizon.

4. SIMULATIONS

In this section, we present the simulation results to validate the proposed controller. The simulations show a Pioneer 3-DX tracking a trajectory generated by a suitable motion planning algorithm to push an object into the desired configuration (position plus orientation). To simulate the system, we use the V-REP simulator controlled by a MATLAB script which handles the data collection and the controller calculations [2].

4.1 Planner

The final goal is to perform a pushing manipulation task with a mobile robot. The considered object is a box sliding on the floor. The robot interacts with the box through the frontal bumper. Lynch [1992] provides a way to compute the set of velocities that can be used to push a polygonal object without breaking the contact configuration. Using only velocities within the set defined by Lynch [1992] guarantees the controllability of the system, as proved by Lynch and Mason [1996], and it is possible to retrieve the sequence of pushing manoeuvres required to bring the object into the desired configuration.

The planning algorithm proposed in this paper applies the Open Motion Planning Library by Şucan et al. [2012] to the problem of finding the sequence of pushing manoeuvres. The presence of static obstacles is addressed as well. The resulting list of pushing manoeuvres is elaborated by a kinematic predictor that converts the list of velocities into reference positions for the robot. The predictor also inserts the procedures required by the robot to circumnavigate the object and changing the pushing side when required. The resulting trajectory is used to perform all the simulations carried out below.

4.2 Case studies

To compute the discretized model of the system (7), the following parameters are retrieved from the datasheet of the physical robot: $L = 0.17$ m, $R = 0.095$ m, $m = 17$ kg, $m_c = 16$ kg, $I_r = 0.1307$ kgm^2, $I_w = 0.0051$ kgm^2, and $d = 0.15$ m. The matrices of the functional cost (4) are chosen to reduce the local error along the Y_R component in Σ_R. This choice implies that the robot moves to initially reduce, as much as possible, the lateral error from the target, and orientation is only adjusted afterwards. Then, by experimentally tuning, the chosen gains are $Q = \text{diag}([2.5\ 5\ 0.25\ 0.05\ 0.05])$, $P = 4Q$, while W and W_N are zero matrices. To compute $G(z)$, instead, the parameters are experimentally tuned as $x_{lim} = y_{lim} = 10$ m, $\Omega_{lim} = 4\pi$ rad/s, $h = \Delta = 0.12$ m, and $\mu = 0.5$. The applied prediction horizon is set as $N = 15$ with control horizon set as $M = 6$. The time interval between the predicted instants is $T_s = 0.1s$. The simulations are performed on a standard PC, with Intel Core i7-4510U CPU, on which it is installed MATLAB R2018b and V-REP 3.5.0. A suitable software developed in MATLAB is in charge of elaborating the measures acquired from V-REP, solve the NMPC algorithm and

[2] A video of the simulations is available at the following link: `https://goo.gl/MhqV6i`

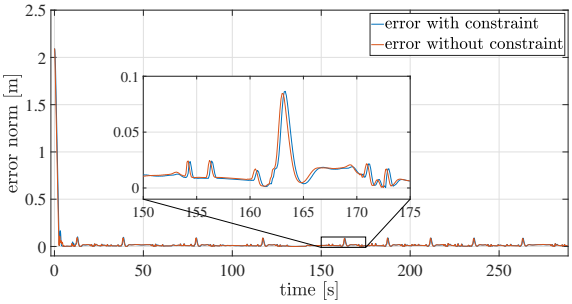


Fig. 4. Position error with and without constraint.

send the actuation command to V-REP, which is thus employed as a dynamic simulator and not as just a visualizer.

Three case studies are addressed in the following. The former is necessary to test the NMPC approach for tracking the trajectory obtained from Section 4.1 with a mobile robot. The second case study shows the pushing manipulation on the same trajectory, while the latter case study includes some parametric uncertainties and discrepancy between the simulated robot in V-REP and the parameters employed for the NMPC algorithm in MATLAB.

Case study I. In this case study, the robot must follow a given trajectory without slipping. First, the robot moves towards the initial point of the trajectory until it reaches a 0.05 m radius around the point. Then, the robot starts following the trajectory. To validate the constraint, we executed a simulation with the wheel slip avoidance constraints (20)—(21) active, and a simulation without the constraint active. Figure 4 shows the tracking errors for both the simulations. Figure 5 shows the robot tracking the trajectory with the constraint active: red blocks represent obstacles to be avoided. The trajectory, obtained exploiting the planner introduced in Section 4.1, starts in the bottom-right part of the plot (position $(0, -3)$), and is composed of different portions: arc-shaped portions are introduced to let the robot change its orientation. It is possible to notice that the robot follows the trajectory in a very precise manner. Figure 6 and 7 show the difference between the surface velocity of each wheel and the longitudinal velocity of its axle for the first 13 s of simulation: when this difference is non-zero, then wheel slip is happening. The plots clearly show that the application of the constraints significantly reduces the presence of wheel slip.

Case study II. In this second case study, pushing manipulation is addressed. The box to be pushed is large $0.25 \times 0.25 \times 0.25$ m, and it weights 3 kg. The presence of the box sliding on the floor acts as an unmodeled disturbance to the system. The robot behaves as discussed in case study I, following a trajectory defined according to the planner introduced in Section 4.1: it moves to the starting point, and then continues its motion, tracking the trajectory. Figure 8 shows the position error of the robot during the simulation. The robot can track the trajectory regardless of the presence of the box. In Figure 9 we observe the longitudinal velocity difference of each wheel across the whole simulation. The differences remain bounded in spite of the presence of the box.

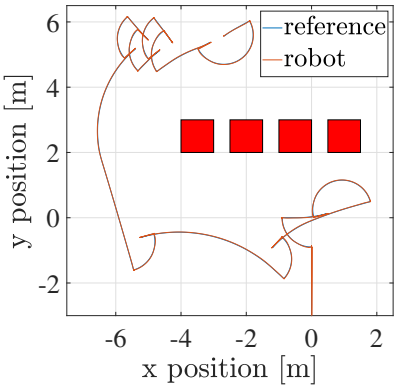


Fig. 5. Robot trajectory and reference: red blocks represent obstacles.

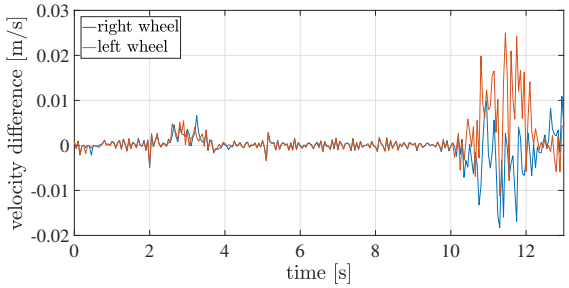


Fig. 6. Longitudinal velocity difference with constraint: non-zero values imply the presence of wheel slip.

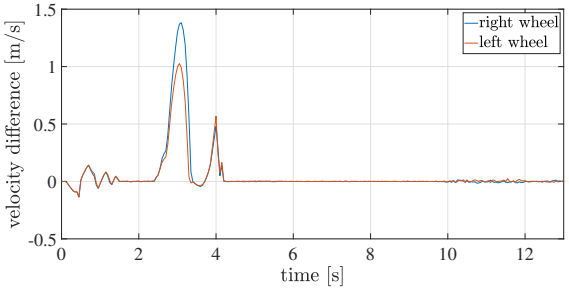


Fig. 7. Longitudinal velocity difference without constraint: non-zero values imply the presence of wheel slip.

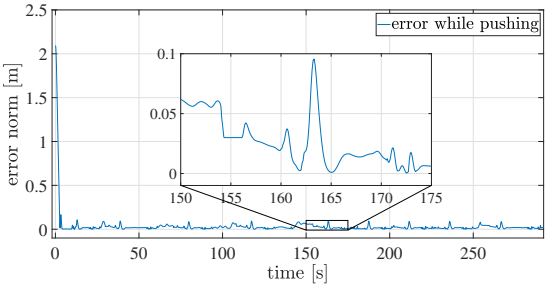


Fig. 8. Position error for the pushing manipulation,

Case study III. In a real environment, it is very likely to have inaccuracies in parameters, especially the friction coefficient μ. To ensure the controller is robust to uncertainty regarding this parameter, we conducted a simulation where the value of the friction coefficient for the wheels in the controller's parameters (implemented in MATLAB) is set to $\mu = 0.7$. As per the other simulations,

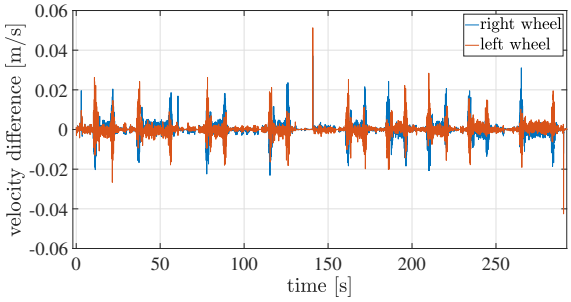


Fig. 9. Longitudinal velocity difference with the box.

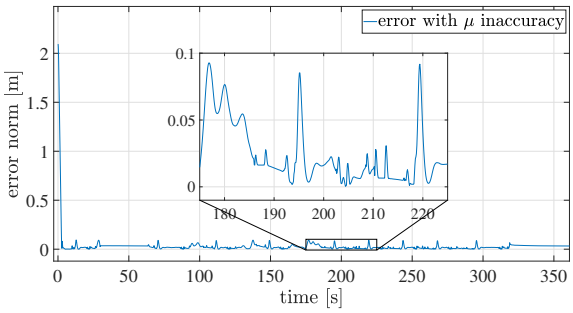


Fig. 10. Position error for the pushing manipulation with inaccurate μ.

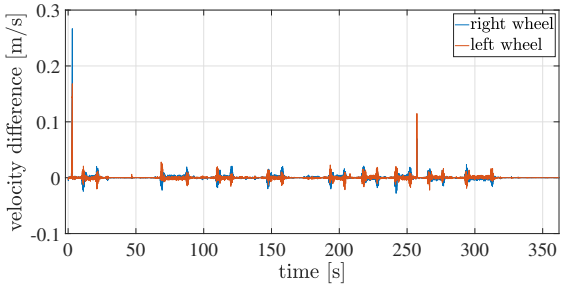


Fig. 11. Longitudinal velocity difference with inaccurate μ.

the robot moves to track the assigned trajectory. Figure 10 shows the tracking performance of the robot subject to disturbance. From Fig. 11 we can conclude that the controller is robust to inaccuracies in the friction parameter.

5. CONCLUSION AND FUTURE WORK

In this paper, we introduced a novel methodology to guarantee wheel slip avoidance. Pushing manipulation is addressed as a case study. The proposed method exploits an NMPC to include both the robot dynamics and the constraints to avoid wheel slipping. For validation purposes, we implemented the proposed method in a dynamic simulation environment, where a mobile robot was controlled to push a box on a flat floor.

The result proposed in this paper represents a first attempt towards the application of nonprehensile manipulation strategies for single and multiple mobile robots. Future works will aim at augmenting the considered NMPC by explicitly including the contact constraints between the box and the frontal bumper. Multi-agent strategies will be addressed as well.

REFERENCES

R. Dhaouadi and A. A. Hatab. Dynamic modelling of differential-drive mobile robots using lagrange and newton-euler methodologies: A unified framework. *Advances in Robotics and Automation*, 2(2):1–7, 2013.

M. A. Golkar, S. T. Namin, and H. Aminaiee. Fuzzy controller for cooperative object pushing with variable line contact. In *IEEE International Conference on Mechatronics (ICM)*, pages 1–6, 2009.

L. Grüne and J. Pannek. *Nonlinear Model Predictive Control: Theory and Algorithms*. 01 2017. ISBN 978-3-319-46024-6.

T. Igarashi, Y. Kamiyama, and M. Inami. A dipole field for object delivery by pushing on a flat surface. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 5114–5119, May 2010.

Y.-H. Kim and D. A. Shell. Using a compliant, unactuated tail to manipulate objects. *IEEE Robotics and Automation Letters*, 2(1):223–230, 2017.

P. Kolhe, N. Dantam, and M. Stilman. Dynamic pushing strategies for dynamically stable mobile manipulators. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3745–3750, 2010.

K. Kovac, I. Zivkovic, and B. D. Basic. Simulation of multi-robot reinforcement learning for box-pushing problem. In *Proceedings of the 12th IEEE Mediterranean Electrotechnical Conference (IEEE Cat. No.04CH37521)*, volume 2, pages 603–606 Vol.2, May 2004.

S. Krivic, E. Ugur, and J. Piater. A robust pushing skill for object delivery between obstacles. In *IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1184–1189, Aug 2016.

K. M. Lynch. The mechanics of fine manipulation by pushing. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2269–2276, 12 1992.

K. M. Lynch and M. T. Mason. Stable pushing: Mechanics, controllability, and planning. *The International Journal of Robotics Research*, 15(6):533–556, 1996.

T. Maneewarn and P. Detudom. Mechanics of cooperative nonprehensile pulling by multiple robots. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2004–2009, 2005.

R. M. Murray, S. S. Sastry, and L. Zexiang. *A Mathematical Introduction to Robotic Manipulation*. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1994. ISBN 0849379814.

F. Ruggiero, V. Lippiello, and B. Siciliano. Nonprehensile dynamic manipulation: A survey. *IEEE Robotics and Automation Letters*, 3(3):1711–1718, 2018.

I. A. Şucan, M. Moll, and L. E. Kavraki. The Open Motion Planning Library. *IEEE Robotics and Automation Magazine*, 19(4):72–82, December 2012.

Z. Wang and M. Schwager. Force-amplifying n-robot transport system (force-ANTS) for cooperative planar manipulation without communication. *The International Journal of Robotics Research*, 35(13):1564–1586, 2016.

A. Yamashita, T. Arai, J. Ota, and H. Asama. Motion planning of multiple mobile robots for cooperative manipulation and transportation. *IEEE Transactions on Robotics and Automation*, 19(2):223–237, 2003.