Unwieldy Object Delivery with Nonholonomic Mobile Base
A Free Pushing Approach

Tang, Yujie; Wisse, Martijn; Pan, Wei

DOI
10.1109/LRA.2024.3455855

Publication date
2024

Document Version
Final published version

Published in
IEEE Robotics and Automation Letters

Unwieldy Object Delivery With Nonholonomic Mobile Base: A Free Pushing Approach

Yujie Tang, Martijn Wisse, *Member, IEEE*, and Wei Pan, *Member, IEEE*

***Abstract*—This letter explores the problem of delivering unwieldy objects using nonholonomic mobile bases. We propose a new approach called *free pushing* to address this challenge. Unlike previous *stable pushing* methods which maintain a stiff robot-object contact, our approach allows the robot to maneuver around the object while pushing it. It aims to execute continuous pushes without losing contact for improved pushing maneuverability. Additionally, to ensure the feasibility of the planned pushes, a robot-object contact model is developed to account for the shape and kinematics of the robot in pushing modeling and planning. A Model Predictive Controller solves the pushing planning problem in real time. Experimental results show that the proposed method achieves an average success rate of 83% with an accuracy of 0.085 m when pushing to the selected goals. Compared to the baselines, this approach improves the agility and efficiency of mobile pushers. Furthermore, it is robust in achieving the task while tolerating modeling errors.**

***Index Terms*—Contact modeling, manipulation planning, motion control.**

I. Introduction

MOBILE manipulators hold great potential for practical applications, such as logistics operations, where they may need to move objects that are unwieldy, either too heavy or too large, for their manipulator arm(s) to grasp. A viable solution is to equip the robot with the ability to push the object, an essential motion primitive for handling objects of varying sizes. In this letter, we concentrate on pushing using the basic mobile base [1], [2], [3], in contrast to recent studies that focus on pushing with a manipulator's end effector [4].

In the field of pushing with robot arms, *stable pushing* approach is widely used. It assumes a stiff contact while pushing [1], [3], [5]. However, for this assumption to be valid, the contact forces imposed by the robot on the object must remain within the friction cone, causing nonholonomic constraint on the motion of the pushed object [6], [7]. As a result, the mobility of the pushed object is limited. Especially when pushed with mobile base pushers, which are mostly nonholonomic wheeled robots, stable pushing would further undermine the maneuverability of the robot-object system. In the case of a cylinder-shaped nonholonomic robot, stable pushing only allows it to push the object straight forward or rotate with a fixed turning radius [8].

To improve pushing agility, pushing planners which allow for relative sliding between the robot and object are proposed [9], [10], [11], [12]. In these methods, the robot-object contact is switched between sticking, right-slide, and left-slide modes to provide the required pushing force on the object. But these pushing motions are planned under the assumption that the robot can freely access any point on the object, which may be true for a robot arm but not necessarily for a nonholonomic robot. For example, differential-drive robots can not move sideways to reach the planned contact point. Executing the planned pushes is always difficult on a real robot platform due to the limitations of the robot's kinematics [9].

This letter aims to develop a maneuverable push approach that allows the transportation of objects using powerful mobile base pushers while considering the feasibility of the plans. The proposed approach allows the object to slide relative to the pusher, hence called *free pushing*, in contrast to the *stable pushing* methods. To ensure the feasibility of the planned pushes, planning is conducted for the motion of the robot pusher and the resulting object movement. It is achieved through the development of a robot-object contact model by modeling the robot and the object as a unified multibody system employing Differential Algebraic Equations (DAE) such that the motion of the object relative to the robot can be predicted. Unlike existing pushing models, the relative motion is modeled with careful consideration of the robot's shape and kinematics. To the best of our knowledge, this is the first letter that addresses the shape and kinematics of the robot in pushing modeling and planning. A Model Predictive Controller (MPC) solves the Push Planning Problem in real time.

The contributions of our letter can be summarized as follows:

- We propose a free pushing approach for nonholonomic mobile robot which enables the robot to maneuver around the object while pushing it, allowing for improved pushing maneuverability compared to the stable pushing counterpart.
- We develop a robot-object contact model that takes into account the robot shape and kinematics in pushing modeling and planning, ensuring efficient continuous pushing and the feasibility of the planned pushes.

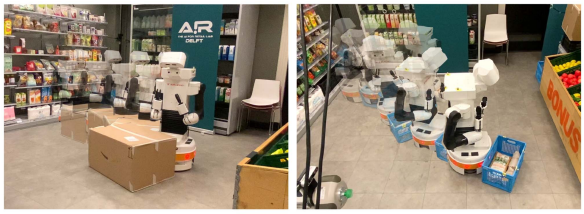
Received 15 May 2024; accepted 26 August 2024. Date of publication 6 September 2024; date of current version 16 September 2024. This article was recommended for publication by Associate Editor F. wyffels and Editor J. Borras Sol upon evaluation of the reviewers' comments. This work was supported by the China Scholarship Council (CSC) under Grant 202006890020. *(Corresponding author: Wei Pan.)*

Yujie Tang and Martijn Wisse are with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, Netherlands (e-mail: y.tang-6@tudelft.nl; m.wisse@tudelft.nl).

Wei Pan is with the Department of Computer Science, The University of Manchester, M13 9PL Manchester, U.K. (e-mail: wei.pan@tudelft.nl).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2024.3455855, provided by the authors.

Digital Object Identifier 10.1109/LRA.2024.3455855



(a) (b)

Fig. 1. Pushing (a) a $0.565 \times 0.755 \times 0.425$ m^3 sized package, and (b) an 8 kg weighed basket, with a differential-drive mobile base. Tiago, a mobile service robot from PAL Robotics [13], is used in this work. Transparency in the image indicates the movements.

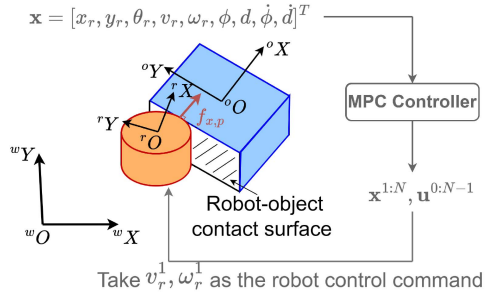


Fig. 2. The configuration of the coordinates and frames, as well as the demonstration of the control framework.

- We evaluate the proposed method through real-world experiments using Tiago (Fig. 1), a mobile service robot, demonstrating an average success rate of 83% with an accuracy of 0.085 m.

II. Related Work

Pushing manipulation has garnered increasing interest because of its ability to handle objects of various sizes and shapes. However, modeling and controlling pushing present challenges stemming from its discontinuity feature, characterized by 1) transitions between different contact modes (sliding up, sticking, and sliding down) caused by friction, and 2) sudden changes in system dynamics upon contact.

Various solutions have been proposed in the literature to address the complex pushing problem posed by its discontinuous feature. One approach uses a hybrid model to represent pushing dynamics in different contact modes [11] and employs a mixed-integer program that optimizes both the discrete contact mode sequence and continuous control actions. On the contrary, [12], [14] represents contact as a complementarity constraint, implicitly planning for contact modes. However, solving a mixed-integer optimization problem or finding a solution for trajectory optimization with complementarity constraints are computationally challenging, especially as the size of the discrete decision variables increases. Therefore, continuous relaxations are often used to simplify the problem. For example, [11] plans continuous pushes without losing contact with the object, while [14] assumes frictionless contact between the robot and the object.

In addition to using continuous approximations, it is recognized that even accurate analytical dynamics models for pushing are inherently unstable because physics parameters such as inertia and friction can only be approximated [15]. Furthermore, the friction parameters may gradually vary over large surfaces, which is hard to be modeled [16]. Thus, simplified models are commonly used instead of looking for an accurate dynamics model, accompanied by real-time control strategies to fill the gaps in model simplification. For instance, ellipsoidal approximation of the limit surface together with the quasi-static interaction assumption have been widely used in robot-pushing applications [7], [10], [17]. This approximation is relatively accurate in determining whether slippage will occur [18], but it cannot quantify the relative motion between the robot and the object. In contrast, [19] uses a simplified object dynamics model to predict the relative motion. While it plans for the optimal contact force, contact point, and robot control input using two separate model predictive control frameworks, it does not guarantee that the planned contact points are reachable.

This letter presents a simplified pushing dynamics model to predict relative robot-object motion and a real-time pushing planner that plans feasible robot pushes. Pushing with robots of different sizes results in distinct object motions and contact point trajectories. Previous work, such as [20], [21], has studied high-level path planning for large-sized disk-shaped robots and objects, but has not provided a low-level control approach to achieve the planned path. As [19] concluded, following the pushing trajectory is equivalent to following the contact point sequence. However, the nonholonomic constraint of a differential-drive robot limits its ability to reach planned contact points.

Therefore, planning achievable contact point sequences is critical to the success of push manipulation. To address this challenge, we propose a pushing planning approach that ensures the plan's feasibility by accounting for the shape and kinematics of the robot.

III. Preliminaries

We consider a pushing planning problem where a cylinder-shaped differential-drive robot pushes a rectangular object toward a goal position. Fig. 2 provides a visualization of the problem setup. The pushing system consists of the cylinder-shaped robot, which can be controlled directly, and the rectangular object, which moves in response to the contact forces. We assume that the contact between the robot and the object is frictionless while using a continuous friction model for the contact between the object and the ground. In the following, we will introduce the friction-less contact model and continuous friction model respectively. Before going into details, we first provide the definitions for the variables used throughout the letter.

- $\mathbf{x}_\mathrm{r}$, the robot state $[x_\mathrm{r}, y_\mathrm{r}, \theta_\mathrm{r}]^\mathrm{T}$ in the world frame;
- $\mathbf{x}_\mathrm{o}$, the object state $[x_\mathrm{o}, y_\mathrm{o}, \theta_\mathrm{o}]^\mathrm{T}$ in the world frame;
- ϕ, the angle between the object and the robot frame as shown in Fig. 3(a)
- d, the y coordinate of robot center in the object frame;
- v_r, ω_r, the linear and angular velocities of the robot;
- a_r, ξ_r, the linear and angular accelerations of the robot;

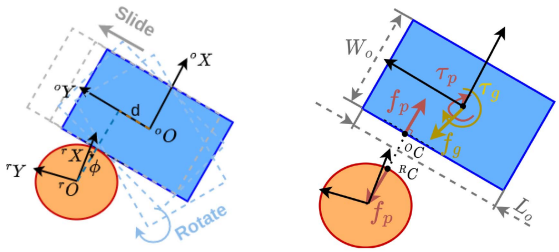


Fig. 3. The robot and the object are assumed to be connected with a virtual sliding joint around the contour of the mobile base. (a) Shows the possible robot-object relative motion which not only includes sliding but also rotating around the contact point. In (b), the robot and the object are taken as two free bodies where the contact joint between them is cut. Then the Newton-Euler equations of motion of the object can be achieved where the joint is "glued" with the constraint equation.

- ${}^{\mathcal{O}}f_{\mathrm{x,p}}$, the x-component of the push force in the object frame;[1]
- $\mathbf{w}_{\mathrm{g}} = [f_{\mathrm{x,g}}, f_{\mathrm{y,g}}, \tau_{\mathrm{g}}]^{\mathrm{T}}$ is the friction wrench exerted by the ground on the object, where $f_{\mathrm{x,g}}, f_{\mathrm{y,g}}$ are the x and y component of the friction force, τ_{g} is the friction torch;
- $\tilde{\mathbf{w}}_{\mathrm{g}} = [\tilde{f}_{\mathrm{x,g}}, \tilde{f}_{\mathrm{y,g}}, \tilde{\tau}_{\mathrm{g}}]^{\mathrm{T}}$, the simplified friction wrench exerted by the ground to the object;
- $M_{\mathrm{o}} = \mathrm{diag}[m_{\mathrm{o}}, m_{\mathrm{o}}, I_{\mathrm{o}}]$, the inertia matrix of the object, where $m_{\mathrm{o}}, I_{\mathrm{o}}$ denote the mass and moment of inertia of the object;
- $W_{\mathrm{o}}, L_{\mathrm{o}}$, width and length of the object;
- r_{r}, radius of the robot.

A. Friction-Less Robot-Object Contact Model

With a point contact between the robot and the object, the object can rotate around that contact point or slide along the contact surface, which can be seen in Fig. 3(a).

Instead of dividing the contact modes into $\{sticking, right-sliding, left-sliding, separation\}$ states [10], [22], [23], we use a frictionless contact assumption to simplify the problem such that there is a continuous sliding contact. Since friction forces only play a minor role in our pushing tasks, this assumption helps reduce computational time at the expense of a slight loss in physical accuracy [14].

In addition, we introduce contact constraints to ensure continuous contact between the robot and the object, avoiding separation contact modes during the pushing process. Specifically, these constraints require that (1a) the robot does not move backward, (1b) the robot does not decelerate abruptly at a rate greater than the object's deceleration due to friction with the ground, (1c) the object remains in front of the robot, and (1d) the robot maintains contact with the object at the same plane. The contact constraints are summarized as:

$$v_{\mathrm{r}} \geq 0 \tag{1a}$$

$$|a_{\mathrm{r}}| < a_{\mathrm{r,\,max}} \tag{1b}$$

$$-90^{\circ} < \phi < 90^{\circ} \tag{1c}$$

$$-\frac{1}{2}L_o < d < \frac{1}{2}L_o \tag{1d}$$

where $a_{\mathrm{r,\,max}}$ is the maximum acceleration of the robot.

B. Continuous Object-Ground Contact Friction Model

While the contact between the robot and the object is assumed to be frictionless in our pushing planning problem, we model the contact between the object and the floor using a continuous friction model. In most previous studies on pushing planning, the Coulomb friction model has been used, and the friction cone has been employed to control the transition between different contact modes. However, the Coulomb model is discontinuous at zero relative tangential contact velocities, which can lead to instability in the numerical simulation of the sliding motion [24]. This characteristic makes the design of controllers more complicated.

Thus we point out the key modeling decision in this letter which is to use a continuous friction model instead of the traditional Coulomb friction model [25]. This choice enables us to model the contact dynamics in a continuous manner, which streamlines the controller design process. To obtain the continuous friction model, we mathematically approximate the Coulomb model using a Sigmoid function. It should be noted that we assume a constant coefficient of friction, μ_{g}, for the friction interaction between the object and the ground at the contact surface. The friction force at position ${}^{\mathcal{W}}\mathrm{p}$ is expressed as

$$\mathbf{f}_{\mathrm{g,p}}({}^{\mathcal{W}}\mathbf{p}) = -\mu_{\mathrm{g}} \cdot (\gamma_1 \mathrm{sig}(\mathbf{v}_{\mathrm{p}}) - [\gamma_2, \gamma_2]^{\mathrm{T}}) \cdot f_{\mathrm{n}} \tag{2}$$

where $\gamma_1 = 2, \gamma_2 = 1$ are scaling factors, f_{n} is the normal force at ${}^{\mathcal{W}}\mathrm{p}$, $\mathbf{v}_{\mathrm{p}} = [\dot{x}_{\mathrm{p}}, \dot{y}_{\mathrm{p}}]^{\mathrm{T}}$ the velocity of ${}^{\mathcal{W}}\mathrm{p}$ expressed in the world frame, and $\mathrm{sig}(\cdot)$ is the sigmoid function that operates element-wise on the input vector, such that $\mathrm{sig}(\mathbf{x})$ yields a vector $\mathbf{y}$, where each element y_i is computed as $y_i = \frac{1}{1+\exp(-(x_i))}$, for $i = 1, 2, \ldots, n$, where n is the length of the input vector $\mathbf{x}$.

IV. Modelling

In this section, we present the dynamics model for the pushing system. First, we introduce the model for the differential-drive robot pusher. Then we will explain the contact transition dynamics between the robot and the object such that the physical interaction is modeled with a 2D (i.e., top-view) constrained multi-body system.

A. Robot Dynamics Model

Given the full robot state $[x_{\mathrm{r}}, y_{\mathrm{r}}, \theta_{\mathrm{r}}, v_{\mathrm{r}}, \omega_{\mathrm{r}}]^{\top}$ and its control input $\mathbf{u}_{\mathrm{r}} = [a_{\mathrm{r}}, \xi_{\mathrm{r}}]^{\top} \in \mathbb{R}^2$, the dynamics model of robot can be written as:

$$\begin{bmatrix} \dot{x}_{\mathrm{r}} \\ \dot{y}_{\mathrm{r}} \\ \dot{\theta}_{\mathrm{r}} \\ \dot{v}_{\mathrm{r}} \\ \dot{\omega}_{\mathrm{r}} \end{bmatrix} = \begin{bmatrix} v_{\mathrm{r}}\cos(\theta_{\mathrm{r}}) \\ v_{\mathrm{r}}\sin(\theta_{\mathrm{r}}) \\ \omega_{\mathrm{r}} \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ a_{\mathrm{r}} \\ \xi_{\mathrm{r}} \end{bmatrix} \tag{3}$$

[1]The superscript $\mathcal{R}$ denotes the robot frame. The object and the world frames are represented as $\mathcal{O}$ and $\mathcal{W}$. $\mathcal{W}$ is always omitted for simplicity.

B. Dynamics Model of the Pushing System

Different from the widely-used point contact model [10], [22], [23] where the robot is a pole and its shape is ignored, we model the relative motion between the robot and the object with careful consideration of the shape of the robot. Since we constrain that the contact between the robot and the object is carefully maintained, the robot-object system is modeled as two rigid bodies connected by an idealized joint. Furthermore, we assume that the mobile robot is controlled in a closed loop to achieve the desired movement, so its motion will not be influenced by the reaction force exerted by the object.

For the pushed object, its motion is a result of the contact interaction with the robot, as shown in Fig. 3. The dynamics of the object can be achieved by a combination of the Newton-Euler equations of motion and a contact constraint equation in the form of Differential Algebraic Equations.

The Newton Euler equations of motion for the object are

$$\begin{cases} \cos(\theta_o)\,{}^{\mathcal{O}}f_{\mathrm{x,p}} + f_{\mathrm{x,g}} = m_{\mathrm{o}}\ddot{x}_{\mathrm{o}} \\ \sin(\theta_o)\,{}^{\mathcal{O}}f_{\mathrm{x,p}} + f_{\mathrm{y,g}} = m_{\mathrm{o}}\ddot{y}_{\mathrm{o}} \\ -d\,{}^{\mathcal{O}}f_{\mathrm{x,p}} + \tau_{\mathrm{g}} = I_{\mathrm{o}}\ddot{\theta}_{\mathrm{o}} \end{cases} \tag{4}$$

where $d = -\sin(\theta_o) * (x_r - x_o) + \cos(\theta_o) * (y_r - y_o)$ is the position of the contact point in y direction of the Object frame.

To solve for the three unknown accelerations of the object as well as the push force in (4), we need one more equation. Since the successive interaction restricts the separate movement of the two bodies at the contact point, we have the contact constraint:

$${}^{\mathcal{O}}x_{\mathrm{r}} = \cos(\theta_{\mathrm{o}})(x_{\mathrm{r}} - x_{\mathrm{o}}) + \sin(\theta_{\mathrm{o}})(y_{\mathrm{r}} - y_{\mathrm{o}}) = -r_{\mathrm{r}} - \frac{1}{2}W_{\mathrm{o}} \tag{5}$$

By differentiating (5) twice with respect to time, we achieve the constraint on the object accelerations:

$$C_{\mathbf{x}}\ddot{\mathbf{x}}_{\mathrm{r}} + g_{\mathbf{x}} = 0 \tag{6}$$

with $g_{\mathbf{x}} = -\cos(\theta_{\mathrm{o}})\dot{\theta}_{\mathrm{o}}\dot{\theta}_{\mathrm{o}}(x_{\mathrm{r}} - x_{\mathrm{o}}) + \cos(\theta_{\mathrm{o}})\ddot{x}_{\mathrm{r}} + \sin(\theta_{\mathrm{o}})\ddot{y}_{\mathrm{r}} - 2\sin(\theta_{\mathrm{o}})\dot{\theta}_{\mathrm{o}}(\dot{x}_{\mathrm{r}} - \dot{x}_{\mathrm{o}}) - \sin(\theta_{\mathrm{o}})\dot{\theta}_{\mathrm{o}}\dot{\theta}_{\mathrm{o}}(y_{\mathrm{r}} - y_{\mathrm{o}}) + 2\cos(\theta_{\mathrm{o}})\dot{\theta}_{\mathrm{o}}(\dot{y}_{\mathrm{r}} - \dot{y}_{\mathrm{o}})$, and $C_{\mathbf{x}} = [-\cos(\theta_{\mathrm{o}}),\ -\sin(\theta_{\mathrm{o}}),\ -\sin(\theta_{\mathrm{o}})(x_{\mathrm{r}} - x_{\mathrm{o}}) + \cos(\theta_{\mathrm{o}})(y_{\mathrm{r}} - y_{\mathrm{o}})]$.

Combining the equations of motion (4) and the contact constraint equation (6) leads to the full set of DAEs:

$$\begin{bmatrix} M_{\mathrm{o}} & C_{\mathbf{x}}^{\mathrm{T}} \\ C_{\mathbf{x}} & 0 \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{x}}_{\mathrm{o}} \\ {}^{\mathcal{O}}f_{\mathrm{x,p}} \end{bmatrix} = \begin{bmatrix} \mathbf{w}_{\mathrm{g}} \\ g_{\mathbf{x}} \end{bmatrix} \tag{7}$$

where $\mathbf{w}_{\mathrm{g}}$ is calculated by integrating the force over the contact patch A: $\mathbf{f}_{\mathrm{g}} = \int_R -\mu_{\mathrm{g}}(2\mathrm{sig}(\mathbf{v}_{\mathrm{p}}) - [1,1]^{\mathrm{T}})p({}^{\mathcal{O}}\mathbf{p})dA$, $p({}^{\mathcal{O}}\mathbf{p})dA$ is the normal force at ${}^{\mathcal{O}}\mathbf{p}$ under pressure distribution $p(\cdot)$, ${}^{\mathcal{O}}\mathbf{p}$ is the position expressed in the object frame. The corresponding frictional moment is $\tau_{\mathrm{g}} = [{}^{\mathcal{O}}\mathbf{p} \otimes \mathbf{f}_{\mathrm{g}}]_{(3)}$, with $\otimes$ cross product, $[\cdot]_{(3)}$ indicates the third element in the vector.

C. Model Simplification

However, the precision of the double integration depends on the number of segmented contact patches. Integration with more contact patches promises a more accurate approximation to the integral but slow computation. For computational efficiency, we simplified the friction model of the object-ground contact by only summing for the force at the four corners of the polygonal object, instead of integrating over the whole contact region. Assuming a uniform object density when projected onto the horizontal plane, the normal force at four corners is $\frac{1}{4}m_o g$, where g refers to the acceleration of gravity. Using (2), the simplified friction force is ${}^{\mathcal{W}}\tilde{\mathbf{f}}_{\mathrm{g}} = \sum {}^{\mathcal{W}}\mathbf{f}_{\mathrm{g,i}}({}^{\mathcal{O}}\mathbf{p}_{\mathrm{c,i}})$ where $i = \{0,1,2,3\}$ for a rectangle object, ${}^{\mathcal{O}}\mathbf{p}_{\mathrm{c,i}}$ is the coordinate of the corner at contact patch A. The corresponding simplified frictional moment is $\tilde{\tau}_{\mathrm{g}} = \sum[{}^{\mathcal{O}}\mathbf{p}_{\mathrm{c,i}} \otimes {}^{\mathcal{W}}\tilde{\mathbf{f}}_{\mathrm{g,i}}]_{(3)}$.

V. Planning for Robot Pushing

By predicting the motion of the pushed object using (7), we can solve the constrained pushing planning problem using a Model Predictive Controller. It ensures the feasibility of the MPC-planned pushes by planning for both the path of the robot and the motion of the pushed object. First, we will discuss how to reformulate the DAE model to simplify the optimization problem in MPC. Afterward, we will present the Optimal Control Problem (OCP) and the cost function.

A. Dynamics Model in Generalized Coordinates With Implicit Constraints

To solve the pushing control problem in an MPC formulation, a system dynamic model is required. The robot-pusher system is modelled as a constrained multi-body system whose dynamics is derived with a set of equations of motion and an algebraic constraint equation in (7). However, it is known that many numerical discretization schemes, such as the implicit Runge–Kutta (IRK) methods, fail to converge or exhibit an order reduction when applied to DAEs [26]. The algebraic constraints for the system state make the forward prediction problem significantly hard.

Removing the algebraic constraint in (6), obviously can help by making the problem less complex. Thus we reformulate the DAEs to simplify the optimal control problem. By choosing a set of independent generalized coordinates, we derive the system dynamics where the algebraic contact constraint can be imposed implicitly.

The object only has two degrees of freedom with respect to the robot, as shown in Fig. 3(a). Therefore, the independent generalized coordinates are chosen as $\mathbf{q} = [\phi, d]$. With the contact constraint between the robot and the object during pushing, the coordinates of the object in the robot frame can be written as a function of the generalized coordinates and their derivatives:

$$\begin{bmatrix} {}^{\mathcal{R}}x_{\mathrm{o}} \\ {}^{\mathcal{R}}y_{\mathrm{o}} \\ {}^{\mathcal{R}}\theta_{\mathrm{o}} \end{bmatrix} = \begin{bmatrix} \cos(\phi)(W_{\mathrm{o}}/2 + r_{\mathrm{r}}) + d\sin(\phi) \\ \sin(\phi)(W_{\mathrm{o}}/2 + r_{\mathrm{r}}) - d\cos(\phi) \\ \phi \end{bmatrix} \tag{8}$$

We differentiate (8) twice:

$${}^{\mathcal{R}}\dot{\mathbf{x}}_{\mathrm{o}} = T\dot{\mathbf{q}} \tag{9a}$$

$${}^{\mathcal{R}}\ddot{\mathbf{x}}_{\mathrm{o}} = T\ddot{\mathbf{q}} + \mathbf{g} \tag{9b}$$

where ${}^{\mathcal{R}}\ddot{\mathbf{x}}_{\mathrm{o}} = [{}^{\mathcal{R}}\ddot{x}_{\mathrm{o}}, {}^{\mathcal{R}}\ddot{y}_{\mathrm{o}}, {}^{\mathcal{R}}\ddot{\theta}_{\mathrm{o}}]^{\mathrm{T}}$, $\mathbf{g} = [-\cos(\phi)\dot{\phi}^2(w_{\mathrm{o}}/2 + r_{\mathrm{r}}) + \dot{d}\cos(\phi)\dot{\phi} - d\sin(\phi)\dot{\phi}^2 + \dot{d}\cos(\phi)\dot{\phi}, -\sin(\phi)\dot{\phi}^2(w_{\mathrm{o}}/2 + r_{\mathrm{r}}) + \dot{d}\sin(\phi)\dot{\phi} + d\cos(\phi)\dot{\phi}^2 + \dot{d}\sin(\phi)\dot{\phi}, 0]^{\mathrm{T}}$ and

$$T = \begin{bmatrix} -\sin(\phi)(w_{\mathrm{o}}/2 + r_{\mathrm{r}}) + d\cos(\phi) & \sin(\phi) \\ \sin(\phi)(w_{\mathrm{o}}/2 + r_{\mathrm{r}}) + d\sin(\phi) & -\cos(\phi) \\ 1 & 0 \end{bmatrix}.$$

${}^{\mathcal{R}}\ddot{\mathbf{x}}_{\mathrm{o}}$ can be achieved by reformulating the dynamics in (4) in the non-inertia reference frame (the Robot frame):

$$M_{\mathrm{o}}\,{}^{\mathcal{R}}\ddot{\mathbf{x}}_{\mathrm{o}} = {}^{\mathcal{R}}\mathbf{w}_{\mathrm{p}} + {}^{\mathcal{R}}\tilde{\mathbf{w}}_{\mathrm{g}} - {}^{\mathcal{R}}\mathbf{w}_{\mathrm{fic}} \tag{10}$$

where ${}^{\mathcal{R}}\mathbf{w}_{\mathrm{p}} \in \mathbb{R}^3$ denotes the push wrench exerted by the robot, ${}^{\mathcal{R}}\mathbf{w}_{\mathrm{fic}} = 2m_{\mathrm{o}}\dot{\boldsymbol{\theta}}_{\mathrm{r}} \times {}^{\mathcal{W}}\dot{\mathbf{p}}_{\mathrm{r}} + m_{\mathrm{o}}{}^{\mathcal{W}}\ddot{\mathbf{p}}_{\mathrm{r}} + m_{\mathrm{o}}\dot{\boldsymbol{\theta}}_{\mathrm{r}} \times (\dot{\boldsymbol{\theta}}_{\mathrm{r}} \times {}^{\mathcal{W}}\mathbf{p}_{\mathrm{r}}) + I_{\mathrm{o}}\ddot{\boldsymbol{\theta}}_{\mathrm{r}} \times {}^{\mathcal{W}}\mathbf{p}_{\mathrm{r}}$ is the introduced additional fictitious wrenches with $\dot{\boldsymbol{\theta}}_{\mathrm{r}} = [0, 0, \omega_{\mathrm{r}}]$, ${}^{\mathcal{W}}\mathbf{p}_{\mathrm{r}} = [x_{\mathrm{r}}, y_{\mathrm{r}}, 0]$.

To achieve the object dynamics with generalized coordinates, the TMT method proposed in [27], is used with the application of the principle of virtual power. Based on the DAE achieved in Section IV, the TMT method simply derives the unconstrained equations of motion [28].

Since we have introduced the generalized coordinates, we also have a new set of corresponding generalized force $\mathbf{f}_q$. Given the wrench of generalized force as $\mathbf{w}_q$, the total virtual power of ${}^{\mathcal{R}}\tilde{\mathbf{w}}_{\mathrm{g}}$, ${}^{\mathcal{R}}\mathbf{w}_{\mathrm{fic}}$ and $\mathbf{w}_q$ is

$$\delta P = \delta\,{}^{\mathcal{R}}\dot{\mathbf{x}}_{\mathrm{o}}^{\mathrm{T}}({}^{\mathcal{R}}\tilde{\mathbf{w}}_{\mathrm{g}} - M_{\mathrm{o}}\,{}^{\mathcal{R}}\ddot{\mathbf{x}}_{\mathrm{o}} - {}^{\mathcal{R}}\mathbf{w}_{\mathrm{fic}}) + \delta\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{w}_q \tag{11}$$

The virtual velocities and the accelerations which satisfy the constraints are shown in (9a)–(9b). Substitution in (11) yields the virtual power expressed in generalized coordinates and their derivatives,

$$\delta P = (T\delta\dot{\mathbf{q}})^{\mathrm{T}}({}^{\mathcal{R}}\tilde{\mathbf{w}}_{\mathrm{g}} - M_{\mathrm{o}}(T\ddot{\mathbf{q}} + \mathbf{g}) - {}^{\mathcal{R}}\mathbf{w}_{\mathrm{fic}}) + \delta\dot{\mathbf{q}}^{\mathrm{T}}\mathbf{w}_q \tag{12}$$

Because the system is in dynamic equilibrium where $\delta P = 0$, which results in

$$T^{\mathrm{T}}({}^{\mathcal{R}}\tilde{\mathbf{w}}_{\mathrm{g}} - M_{\mathrm{o}}T\ddot{\mathbf{q}} - M_{\mathrm{o}}\mathbf{g} - {}^{\mathcal{R}}\mathbf{w}_{\mathrm{fic}}) + \mathbf{w}_q = 0 \tag{13}$$

$\mathbf{f}_q$ equals the friction force between robot and object, which is omitted because of the friction-less assumption in Section III-A. Then we have $\mathbf{w}_q = 0$. Rearranging the terms in (13) gives us contact dynamics in terms of generalized coordinates.

$$\ddot{\mathbf{q}} = T^{\mathrm{T}}M_{\mathrm{o}}({}^{\mathcal{R}}\tilde{\mathbf{w}}_{\mathrm{g}} - {}^{\mathcal{R}}\mathbf{w}_{\mathrm{fic}} - M_{\mathrm{o}}\mathbf{g}) \tag{14}$$

Hence, the dynamics of the whole robot-object system is the combination of (3) and (14).

$$\begin{bmatrix} \dot{x}_{\mathrm{r}} \\ \dot{y}_{\mathrm{r}} \\ \dot{\theta}_{\mathrm{r}} \\ \dot{v}_{\mathrm{r}} \\ \dot{\omega}_{\mathrm{r}} \\ \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \begin{bmatrix} v_{\mathrm{r}}\cos(\theta_{\mathrm{r}}) \\ v_{\mathrm{r}}\sin(\theta_{\mathrm{r}}) \\ \omega_{\mathrm{r}} \\ a_{\mathrm{r}} \\ \xi_{\mathrm{r}} \\ \dot{\mathbf{q}} \\ T^{\mathrm{T}}M_{\mathrm{o}}({}^{\mathcal{R}}\tilde{\mathbf{w}}_{\mathrm{g}} - {}^{\mathcal{R}}\mathbf{w}_{\mathrm{fic}} - M_{\mathrm{o}}\mathbf{g}) \end{bmatrix} \tag{15}$$

B. Optimal Control Problem (OCP) Formulation

With the simplified motion model, a desired controller must be able to recover from the perturbation and be fast enough to do re-planning online. Additionally, it is important to obey the contact constraint in (1) so that the derived contact dynamics is applied. To satisfy these requirements, we formulate an OCP where the goal is to minimize the finite-horizon cost-to-go function subject to the constraints and the dynamics of the system at every control cycle.

We first define the system state vector $\mathbf{x}$ and control input $\mathbf{u}$ as $\mathbf{x} = [x_{\mathrm{r}}, y_{\mathrm{r}}, \theta_{\mathrm{r}}, v_{\mathrm{r}}, \omega_{\mathrm{r}}, \phi, d, \dot{\phi}, \dot{d}]^{\mathrm{T}}$ and $\mathbf{u} = [a_{\mathrm{r}}, \xi_{\mathrm{r}}]^{\mathrm{T}}$. Then the cost-to-go for N time steps is set as: $J^{\mathrm{N}}(\mathbf{x}^{\mathrm{N}}) = ({}^{\mathcal{W}}\mathbf{p}_{\mathrm{o}}^{\mathrm{N}} - {}^{\mathcal{W}}\mathbf{p}_{\mathrm{o}}^{G})Q({}^{\mathcal{W}}\mathbf{p}_{\mathrm{o}}^{\mathrm{N}} - {}^{\mathcal{W}}\mathbf{p}_{\mathrm{o}}^{G})^{\mathrm{T}}$ where Q denotes the weight matrix associated with the object state, ${}^{\mathcal{W}}\mathbf{p}_{\mathrm{o}}^{\mathrm{N}} = [{}^{\mathcal{W}}x_{\mathrm{o}}^{\mathrm{N}}, {}^{\mathcal{W}}y_{\mathrm{o}}^{\mathrm{N}}, {}^{\mathcal{W}}\theta_{\mathrm{o}}^{\mathrm{N}}]^{\mathrm{T}}$ is the predicted object pose in the world frame, ${}^{\mathcal{W}}\mathbf{p}_{\mathrm{o}}^{G}$ is the target object pose.

The final OCP is defined to minimize the distance between the object pose and the target pose at the end of the overall horizon, while satisfying the contact constraints in (1), the state constraints and the robot dynamics:

$$\min_{\mathbf{x}^{1:N},\mathbf{u}^{0:N-1}} \quad J^{\mathrm{N}}(\mathbf{x}^{\mathrm{N}}) \tag{16a}$$

$$\text{s.t.} \quad \mathbf{x}^0 = \mathbf{x}^{t_0} \tag{16b}$$

$$\mathbf{x}^t = \mathbf{f}(\mathbf{x}^{t-1}, \mathbf{u}_{\mathrm{r}}^{t-1}) \tag{16c}$$

$$v_{\mathrm{r}}^t > 0 \tag{16d}$$

$$-90^\circ < \phi^t < 90^\circ \tag{16e}$$

$$|a_{\mathrm{r}}^{t-1}| < a_{\mathrm{r,max}}$$

$$\mathbf{u}^{t-1} \in \mathcal{U}_{\mathrm{r}},\ \forall t \in \{1, \ldots, N\} \tag{16f}$$

It should be noted that $\mathbf{x}^t = \mathbf{f}(\mathbf{x}^{t-1}, \mathbf{u}_{\mathrm{r}}^{t-1})$ is the forward model in (15), and $\mathcal{U}_{\mathrm{r}}$ represents the robot's acceleration and angular acceleration limits. Every time step, the OCP is solved online.

VI. Experimental Results

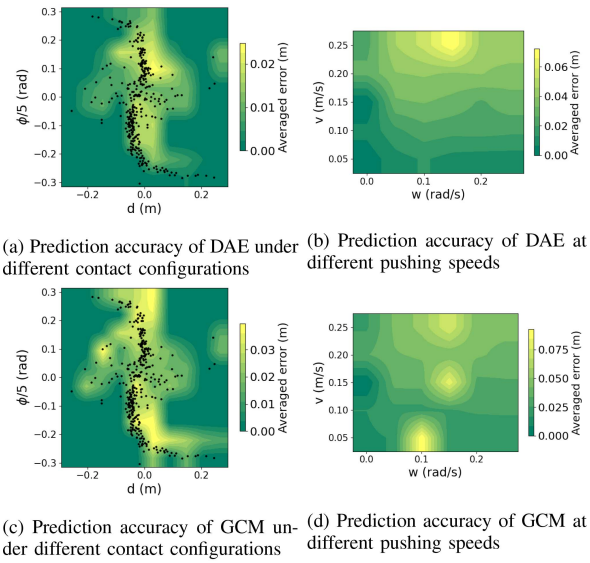
This Section presents an evaluation and validation of the proposed pushing model and controller. The Tiago service robot [13] (see Fig. 1), which has a cylinder-shaped mobile base with a radius of 27cm, was used to push several rectangular boxes with varying physical properties, including mass, size, and texture. The results indicate that our simplified pushing model predicts the object's motion under pushing with an accuracy better than 0.075 m. With this prediction, the proposed controller plans feasible trajectories by adapting the contact with the object while pushing to the goal. Although the pushing success rate varies depending on the target's position, the proposed method can achieve an average success rate of 83% with a pushing accuracy of 0.085 m for the selected goals, distributed equally in the test room. Compared to the stable pushing controller [8], the proposed approach improves the agility and efficiency of mobile manipulators and is robust in achieving the task while tolerating modeling errors.

A. Evaluation of the Contact Model

We evaluate the precision of the contact dynamics model. The Tiago robot is controlled to push a paper box that has a dimension of $0.386 \times 0.585 \times 0.4$ m^3 and a weight of 1.5 kg. The friction coefficient of the ground-box contact is $\mu_{\mathrm{g}} = 0.4$. Two primary sources of prediction errors were identified: the mismatch between the actual shape of the robot and its modeled counterpart and the slippage of the wheeled robot during movement.

We first evaluate the accuracy of the proposed contact model under different contact configurations. To exclude the influence of the robot slippage, we conduct object pushing experiments with a fixed speed as $v_r = 0.1$ m/s, $w_r = 0$ rad/s. 380 trails of

Fig. 4. Boss on the robot.



(a) Prediction accuracy of DAE under different contact configurations

(b) Prediction accuracy of DAE at different pushing speeds

(c) Prediction accuracy of GCM under different contact configurations

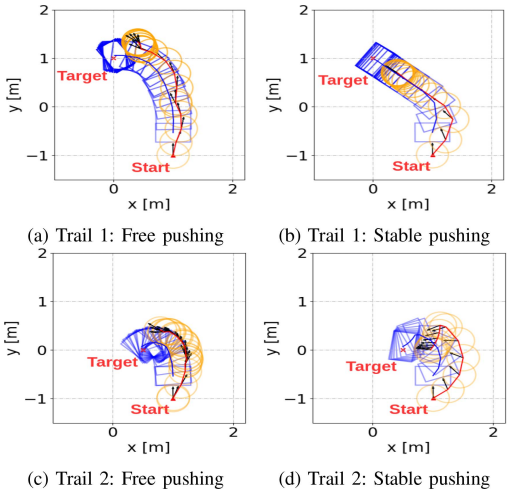
(d) Prediction accuracy of GCM at different pushing speeds

Fig. 5. The prediction accuracy of the DAE model and the simplified GCM evaluated by their averaged prediction error. The black dots in (a), (b) represent the sampled contact configurations. The prediction accuracy is determined by comparing the predicted object position with its actual position recorded by the motion capture system (OptiTrack) after a prediction horizon of 20 timestamps.

the pushing trajectory are collected for 20 timestamps, with a sampling time of $\delta t = 0.1\,s$. The average error of the motion prediction by the DAE model (7) and the simplified Generalized Coordinate Model (GCM, (15)) is presented in Fig. 5(a) and (c), respectively. We found that the overall prediction error was under 0.025 m for the DAE model and 0.04 m for the simplified GCM. However, we observed that the prediction accuracy was lower in cases where the object was in front of the robot at the center ($d = 0, \phi = 0$) due to a mismatch between the real robot and our model. Specifically, there is a boss on the front of the mobile base serving as the charging connector, which causes discontinuous contact dynamics, as shown in Fig. 4.

Furthermore, we evaluate the prediction error under different pushing speed by conducting at least 5 trials of pushing trajectory for each speed configuration, with increments of 0.1\,m/s and 0.05 rd/s for linear and rotation speed, respectively. The results are shown in Fig. 5(b) and (d). It demonstrates an increasing trend in prediction error as pushing speed increases. This could be attributed to the mobile robot's wheel slippage worsening at higher speeds, leading to poorer object motion prediction during pushing. Under all the speed configurations, the DAE model had the highest prediction error of 0.06 m, while the simplified GCM had a prediction error of 0.075 m.

Despite the simplified model's inherent lower prediction accuracy compared to the DAE model, the error resulting from the



(a) Trail 1: Free pushing

(b) Trail 1: Stable pushing

(c) Trail 2: Free pushing

(d) Trail 2: Stable pushing

Fig. 6. Comparison of the pushing performance with a free and stable pushing controller in simulation. The orange circles and blue rectangles represent the trajectory of the robot and object, respectively. The black arrows indicate the direction of the robot. Trails (a), and (b) share the same target of (0, 1), while panels (c) and (d) share the target of (0.5, 0). All panels start with the same initial contact configuration of $d = 0$ and $\phi = 0$.

different contact/speed configurations seems more significant, as shown in Fig. 5. Consequently, a reactive controller is deemed necessary to bridge the gap in the modeling process and tolerate modeling errors.

B. *Validation of the Controller*

1) Simulation Results: Using the same setup as in Section VI-A, we conducted simulations in which the robot pushed the object to different goal positions with varying initial contact configurations. The MPC planner is configured with a prediction horizon of 30 and a sampling time of $\delta t = 0.1\,s$. As a goal-reaching controller, the weight matrix is set as $Q = \mathrm{diag}(1, 1, 0)$. The solver we used is IPOPT.

Three different scenarios were defined, and the proposed free pushing approach is compared with the stable pushing method in [7]. As the simulation results shown in Fig. 6, the robot changes its contact with the object during the pushing process in the free pushing mode, while it maintains a sticking contact with the object in the stable pushing cases. To maintain the sticking contact in stable pushing, the object is modeled as a Dubin's car with its maximum curvature limited [7], as shown in Fig. 6(b). However, it does not consider the kinematic constraints of the nonholonomic robot, which results in infeasible push plans, as seen in Fig. 6(d), where the robot is unable to follow the trajectory to push the object sideways. Furthermore, in cases where the robot and the pushing target are initially positioned on the left of the object, the robot is capable of pushing the object towards the goal by adapting the contact configuration during pushing, as depicted in Fig. 7. But it is not possible with the stable pushing controller where a longer pushing trajectory with a large curvature is required. In conclusion, the free pushing method plans for feasible trajectories for the robot by considering its nonholonomic constraint. It is also more agile and efficient in

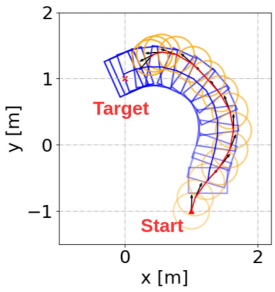


Fig. 7. Free pushing trail 3 with an initial contact configuration of $d = 0.25$ m and $\phi = 0$ and target of (0, 1). The stable pushing is infeasible using the setup in Fig. 6.

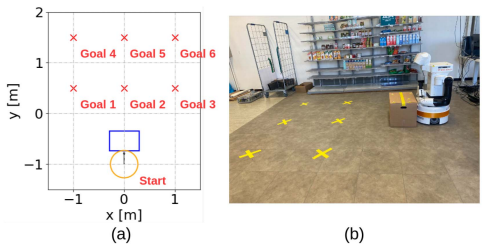


Fig. 8. The selected pushing targets.

TABLE I
PUSHING PERFORMANCE WITH THE REAL ROBOT

Goal	Averaged pushing accuracy (RMSE, m)	Averaged travel distance (m)	Success rate
1	0.074	2.324	0.7
2	0.054	1.427	1
3	0.120	2.510	0.7
4	0.105	3.242	0.8
5	0.064	2.957	1
6	0.098	4.186	0.8

completing pushing tasks due to its ability to change contact points while pushing.

2) Real-World Experimental Results: The proposed controller is tested in physical experiments using (1) the same paper box in Section VI-A and (2) a plastic basket which sizes $0.345 \times 0.522 \times 0.278$ m^3 and weighs 8 kg. The robot and object states are provided by a motion capture system (OptiTrack) and a Kalman filter, which runs at 120 Hz. Control commands are computed on a laptop using our proposed MPC-based method and are sent to the Tiago robot via WiFi and ROS, which runs at 10 Hz. The average solution time of the MPC controller is 60 ms. According to the setup of Tiago, we send the optimized v_{r}^1 and ω_{r}^1 form (16) as its control commands (as demonstrated in Fig. 2).

Ablation experiments are conducted to push the box towards 6 different targets (as shown in Fig. 8), distributed evenly in the motion capture room, for 10 times per target. We defined pushing accuracy as the square root of the average of the squared errors between the goal and the actual final position of the pushed object. We considered cases with pushing accuracy below 0.25 m as failures. The results are summarized in Table I, where we observed an average reaching accuracy below 0.120 m and a delivery success rate higher than 70%. Since the object movement

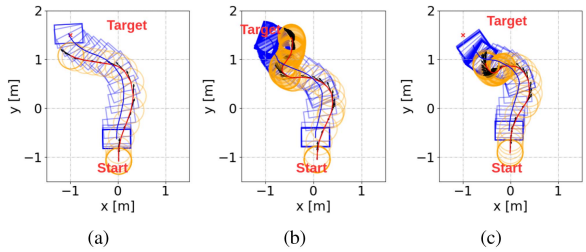


Fig. 9. Pushing trails with the same goal. The object motion is sensitive to the small difference in the control inputs and the initial states. But the robot achieves the push success by continuously adapting the contact configuration.

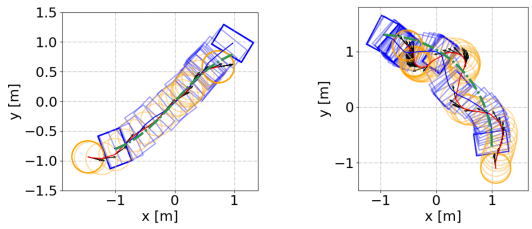


Fig. 10. Tracking an S-curve (left) and a 1/4 circle (right) while engaging in free pushing. The green line represents the reference path.

is very sensitive to the small differences in the contacts and initial states, the robot reacts to unforeseen changes by continuously adapting its contact with the object, resulting in differences in travel distance, even for the same goal. In Fig. 9, we present several repeated pushing trials targeting goal 4. Starting at position $(0, -1)$ with the contact configuration $d = 0, \phi = 0$, the resulting object trajectories varied significantly. Furthermore, as the object got closer to the target, there were instances where the robot's control inputs were too small to articulate the pushing, leading to wheel slippage and failure, as shown in Fig. 9(c). This observation motivates our future research on compliant robot pushing.

As we already discussed in our previous work [8], maintaining contact between the robot and the object is crucial for pushing with nonholonomic robots. Because frequent repositioning actions are time-consuming. Additionally, planning these actions while avoiding collisions with the object simultaneously is challenging. As an improvement of [8], we will not explain it repeatedly due to space limitation. However, we give experiment results of tracking predefined trajectories with the free pushing method (Fig. 10). These results illustrate how the proposed method can continuously push the object along the given path without losing contact. Additionally, the free pushing controller is more maneuverable than [8] to track curly paths without curvature limitation.

Furthermore, we attempted a more challenging task, i.e., pushing a heavy basket filled with random products (Fig. 1(b)). The basket's mass distribution is difficult to measure, complicating accurate system modeling. This test was performed to evaluate the controller's robustness and determine whether it could compensate for small model mismatches. The results of the pushing experiment are shown in Fig. 11, where the controller still successfully achieved the pushing task despite the modeling

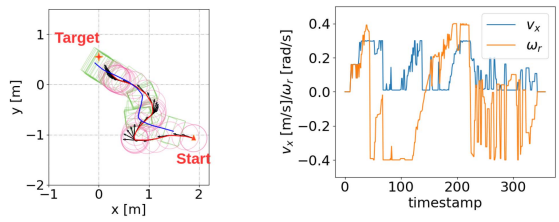


Fig. 11. The robot is able to successfully push a heavy plastic basket to the target position with the tolerance of modeling error. Pictures of the experiment can be seen in Fig. 1(b). (Left) the recorded trajectories, (right) the control commands.

errors. However, it resulted in a longer pushing distance and more changes in the contact configuration. Our results indicate that the mobile base is able to deliver large and heavy objects that cannot be grasped. However, its pushing performance is still limited by motor capabilities and wheel slippage. It is noted that objects that are too heavy may degrade the pushing performance, resulting in jerky motions.

VII. Conclusion

In this letter, we address the challenge of unwieldy object delivery using a differential-drive mobile robot with a free pushing method. Unlike previous approaches focusing on contact point trajectories, we introduce a continuous contact model to predict pusher-slider dynamics and directly plan robot control inputs. This method ensures feasible pushes and minimizes time and distance by continuously pushing the object without relocations. However, it is limited to regular-shaped objects like rectangles.

In contrast, data-driven controllers are suitable for irregular objects but require extensive data collection for tuning. Training a model for each object is impractical. Given that most unwieldy objects in logistics are regular-shaped, we prefer to make an analytical reactive controller, as introduced. An enhancement to the proposed method could involve identifying parameters in the model during the pushing process.

References

[1] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Linear time-varying MPC for nonprehensile object manipulation with a nonholonomic mobile robot," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 11032–11038.

[2] S. Krivic and J. Piater, "Pushing corridors for delivering unknown objects with a mobile robot," *Autono. Robot.*, vol. 43, no. 6, pp. 1435–1452, 2019.

[3] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Characterization of grasp configurations for multi-robot object pushing," in *Proc. IEEE Int. Symp. Multi-Robot. Multi-Agent Syst.*, 2021, pp. 38–46.

[4] J. C. Vaz and P. Oh, "Material handling by humanoid robot while pushing carts using a walking pattern based on capture point," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2020, pp. 9796–9801.

[5] N. Chavan-Dafle and A. Rodriguez, "Stable prehensile pushing: In-hand manipulation with alternating sticking contacts," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2018, pp. 254–261.

[6] P. K. Agarwal, J.-C. Latombe, R. Motwani, and P. Raghavan, "Nonholonomic path planning for pushing a disk among obstacles," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1997, pp. 3124–3129.

[7] J. Zhou, Y. Hou, and M. T. Mason, "Pushing revisited: Differential flatness, trajectory planning, and stabilization," *Int. J. Robot. Res.*, vol. 38, no. 12-13, pp. 1477–1489, 2019.

[8] Y. Tang, H. Zhu, S. Potters, M. Wisse, and W. Pan, "Unwieldy object delivery with nonholonomic mobile base: A stable pushing approach," *IEEE Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7727–7734, Nov. 2023.

[9] E. Arruda, M. J. Mathew, M. Kopicki, M. Mistry, M. Azad, and J. L. Wyatt, "Uncertainty averse pushing with model predictive path integral control," in *Proc. IEEE-RAS 17th Int. Conf. Humanoid Robot.*, 2017, pp. 497–502.

[10] F. R. Hogan, E. R. Grau, and A. Rodriguez, "Reactive planar manipulation with convex hybrid MPC," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2018, pp. 247–253.

[11] F. R. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *Int. J. Robot. Res.*, vol. 39, no. 7, pp. 755–773, 2020.

[12] J. Moura, T. Stouraitis, and S. Vijayakumar, "Non-prehensile planar manipulation via trajectory optimization with complementarity constraints," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 970–976.

[13] J. Pages, L. Marchionni, and F. Ferro, "TIAGo: The modular robot that adapts to different research needs," in *Proc. Int Workshop Robot Modular*, 2016. [Online]. Available: https://www.clawar.org/wp-content/uploads/2016/10/P2.pdf

[14] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, "Contact-implicit trajectory optimization for dynamic object manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2019, pp. 6814–6821.

[15] L. Cong, M. Grner, P. Ruppel, H. Liang, N. Hendrich, and J. Zhang, "Self-adapting recurrent models for object pushing from learning in simulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, 2020, pp. 5304–5310.

[16] A. Zarei Khabjani, H. Karimpour, and M. Keshmiri, "Robotic box pushing under indeterminate anisotropic friction properties," *Int. J. Dyn. Contol*, vol. 9, pp. 872–884, 2021.

[17] N. Doshi, F. R. Hogan, and A. Rodriguez, "Hybrid differential dynamic programming for planar manipulation primitives," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2020, pp. 6759–6765.

[18] A. Fakhari, M. Keshmiri, and M. Keshmiri, "Dynamic modeling and slippage analysis in object manipulation by soft fingers," in *Proc. ASME Int. Mech. Eng. Congr. Expo.*, American Society of Mechanical Engineers, 2014, Art. no. V04AT04A056.

[19] A. Rigo, Y. Chen, S. K. Gupta, and Q. Nguyen, "Contact optimization for non-prehensile loco-manipulation via hierarchical model predictive control," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst,*, 2023, pp. 9945–9951.

[20] M. De Berg and D. H. Gerrits, "Computing push plans for disk-shaped robots," *Int. J. Comput. Geometry Appl.*, vol. 23, no. 1, pp. 29–48, 2013.

[21] D. Nieuwenhuisen, A. F. van der Stappen, and M. H. Overmars, "Path planning for pushing a disk using compliance," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst*, 2005, pp. 714–720.

[22] J. Shi, J. Z. Woodruff, P. B. Umbanhowar, and K. M. Lynch, "Dynamic in-hand sliding manipulation," *IEEE Trans. Robot.*, vol. 33, no. 4, pp. 778–795, Aug. 2017.

[23] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided sampling-based planning for quasistatic dexterous manipulation in 2D," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2021, pp. 6520–6526.

[24] A. Nakashima, Y. Ooka, and Y. Hayakawa, "Contact transition modelling on planar manipulation system with LuGre friction model," in *Proc. IEEE 10th Int. Workshop Robot Motion Contol*, 2015, pp. 300–307.

[25] T. Piatkowski, "GMS friction model approximation," *Mech. Mach. Theory*, vol. 75, pp. 1–11, 2014.

[26] N. Biehn, S. L. Campbell, L. Jay, and T. Westbrook, "Some comments on DAE theory for IRK methods and trajectory optimization," *J. Comput. Appl. Math.*, vol. 120, no. 1-2, pp. 109–131, 2000.

[27] H. Vallery and A. L. Schwab, *Advanced Dynamics*. Delft, Netherlands: Delft University of Technology, 2020.

[28] S. Sadati et al., "AutoTMTDyn: A Matlab software package to drive TMT Lagrange dynamics of series rigid-and continuum-link mechanisms," in *Proc. Workshop Soft Robotic Model. Control*, 2018. [Online]. Available: https://steffen-zschaler.de/assets/pdf/AutoTMTDyn18.pdf