machines

MDPI

Article

Uncertainty-Aware Trajectory Planning and Nonlinear Model Predictive Control for Non-Prehensile Robotic Manipulation

Sara Federico [1], Ciro Natale [1,*], Fabio Ruggiero [2], Mario Selvaggio [2] and Marco Costanzo [1]

[1] Dipartimento di Ingegneria, Università degli Studi della Campania "Luigi Vanvitelli", Via Roma 29, 81031 Aversa, Italy; sara.federico@unicampania.it (S.F.); marco.costanzo@unicampania.it (M.C.)

[2] Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, Università degli Studi di Napoli Federico II, Via Claudio 21, 80125 Napoli, Italy; fabio.ruggiero@unina.it (F.R.); mario.selvaggio@unina.it (M.S.)

* Correspondence: ciro.natale@unicampania.it

Abstract

This paper presents a dual-layered computational framework for the robust trajectory planning and active stabilization of a robotic manipulator transporting a non-fixed payload. The primary challenge addresses the transport of a tray containing multiple objects prone to sliding, exacerbated by significant uncertainties in the system's dynamic parameters, such as objects' mass and inertia. The first contribution is an optimal closed-loop sensitivity-based trajectory planning algorithm that generates energy-efficient paths while minimizing the possibility of object sliding. The second contribution is an active sliding control strategy based on Nonlinear Model Predictive Control (NMPC). This algorithm dynamically adjusts the orientation of the tray, mounted to the robot's end effector, usefully exploiting a dynamic model including inertial forces and gravity to move the object to given positions. Simulation and experimental results demonstrate that the integrated approach allows the robot to set the objects in desired positions with an average steady-state error of 6.3×10^{-3} m across a set of ten experiments, while limiting the sliding to less than 3% of the tray dimensions during successive transportation in face of a 10% uncertainty about objects' masses. The synergy between the uncertainty-aware planner and the NMPC controller supports robust tray-transport tasks in unstructured environments where precise dynamic modeling is unavailable.

Keywords: uncertainty-aware trajectory planning; non-prehensile robotic manipulation; nonlinear model predictive control

1. Introduction

Service robots are increasingly being employed in different fields, from healthcare and hospitality [1] to home care assistance [2], to reduce the operator workload in assisted-care facilities. In particular, robots can dramatically improve healthcare industry efficiency by delivering meals, supplies, medicine, or laboratory materials. In this context, service robots are expected to perform complex manipulation tasks, thus leading to the need for enhanced dexterous manipulation skills. Most service robots are, however, still equipped with basic, two-finger grippers that significantly restrict their manipulation capability to clumsy pick-and-place tasks. To overcome this issue, non-prehensile manipulation techniques enable the use of simpler end effectors to handle objects that are difficult to grasp and manipulate without requiring any form- or force-closure [3]. By avoiding dealing with bilateral constraints, typical of grasping average-based manipulation [4], they additionally

Academic Editor: Firstname Lastname

Received: 29 May 2026
Revised: 1 July 2026
Accepted: 06 July 2026
Published:

 https://doi.org/10.3390/machines1010000

improve the manipulation versatility, allowing robots to perform human-like actions such as pushing [5], throwing [6], and rolling [7]. Despite the increasing research interest in non-prehensile manipulation, current approaches still fall short of reaching human standards, primarily because of oversimplified physical models [8], which lead to poor robustness in uncertain dynamic environments.

Pioneering works [9,10] established the theoretical basis of sensorless sliding-based manipulations and frictional mechanics. Ref. [11] introduced programmable sliding manipulation through controlled motions of a 6 Degrees of Freedom (DoF) plate, showing that periodic tray motions can generate stable asymptotic velocity fields for object positioning and alignment tasks, thus allowing for object manipulation based on intentional sliding dynamics. The work in [8] proposed a Deep Deterministic Policy Gradient Reinforcement Learning framework for efficiently sliding an object on a surface. The developed algorithms enable the relative manipulation of an object as it slides along the surface by controlling the acceleration of a robotic arm, while dynamically estimating the frictional forces during the sliding process. The recent pre-print [12] proposes a simulation framework for dual-arm non-prehensile tray manipulation that integrates RL-learned state transition dynamics as a constraint within a Nonlinear Model Predictive Control (NMPC). Nevertheless, the friction model is an elementary one, and it does not explicitly capture the coupling between rotational and translational friction, as the Limit Surface method combined with the LuGre friction model used in this paper does.

Conversely, exploiting object-tray static friction enables the robot to transport objects without sliding on tray-like end-effectors, increasing their capacity for complex tasks like delivering food to patients in bed [13,14]. Recently, a shared-control teleoperation architecture with sliding prevention was introduced in [13], where adaptive tray tilting proved beneficial in reducing the occurrence of sliding [15]. The problem of planning optimal trajectories enforcing non-sliding constraints for this task has been tackled via quadratic programming [16] and Nonlinear Model Predictive Control [17], showing superior tracking performance. The online planning feasibility of this approach allows easy deployment on robots with significant number of degrees-of-freedom (DOFs) such as mobile manipulators and quadruped robots [18,19].

However, the above-mentioned non-prehensile transportation methods are prone to failures when inaccuracies in friction models or dynamic parameters' uncertainties of the manipulated object arise, especially at significant task speed [20,21]. The robustness of control systems against disturbances and parameter uncertainties has always been a key issue in feedback control, leading to the development of various methods in the literature to address these issues [22–24]. Recent works have addressed uncertainty in non-prehensile manipulation from different perspectives, including task-adaptive planning for pushing [25], learning-based dynamic object transport via uncertainty-aware model predictive schemes [26], and uncertainty-aware policies for mobile non-prehensile manipulation under perception occlusions [27].

Differently from these approaches, this work focuses on a model-based tray-delivery framework exploiting closed-loop-sensitivity-based robust planning for multi-object tray transportation. With this approach, one can plan and execute trajectories that minimize the effect of uncertainties, inherently robustifying the robot behavior [28–32]. The notion of closed-loop sensitivity has emerged as a radically new perspective to manage uncertainties in system parameters without the need to design a dedicated controller [33]. This technique optimizes control-aware trajectories to improve the tracking performance, despite uncertainties in the system–controller pair. This solution has been successfully applied in areas such as quadrotor trajectory tracking [34], energy tank initialization for manipulators [35], and path planning [36]. By exploiting closed-loop sensitivity, it is additionally possible to create tubes that enclose the perturbed system state/input trajectories that can be used to robustify the optimization problem, e.g., by preventing actuator saturation [37]. Only recently has the notion of closed-loop sensitivity been leveraged to robustly handle parametric uncertainties of the manipulated object in a non-prehensile manipulation scenario [21].

In this paper, we propose a unified framework for robot-assisted tray delivery, consisting of a tray setting and a tray transportation phase (see Figure 1), in which a novel active sliding control strategy and a novel optimal trajectory planning algorithm are developed, respectively. The two methods are applied in separate task phases and are not interdependent. The framework is flexible as it allows selecting different algorithms in one phase without affecting the execution of the other phase.

Figure 1. Robot-assisted tray delivery task.

The proposed approach enables a dual-robot system to autonomously arrange the objects directly on the tray while guaranteeing safe transportation conditions. The first robot is in charge of setting the tray by picking and placing the objects on it; during this phase, we actively control the orientation of the second robot holding the tray to generate intentional sliding over the surface and move the object outside the workspace of the first robot. The second robot is then responsible for the delivery phase, which consists of transporting a non-fixed payload while ensuring that no sliding occurs during the motion.

By relying on a purely model-based approach that accounts for a very small number of hyperparameters and does not require long and expensive training phases, we propose a novel physics-based model to analytically describe the sliding of an object over a tray-like surface, by introducing the LuGre friction model [38] along with the ellipsoidal approximation of the Limit Surface [39] to accurately model the frictional forces, which play a crucial role during the task. The proposed model is then used in a nonlinear predictive controller to place the object at a given position by controlled sliding. This model, originally proposed in [40], is known to be capable of reproducing the break-away force phenomenon [41], which contributes to the dependence of the static friction on the rate of the external tangential force. This phenomenon is also relevant in robotic manipulation, especially in the in-hand pivoting of objects through parallel grippers [42].

Moreover, this paper aims to extend the sensitivity-aware robust non-sliding dynamic manipulation planning method presented in [21] to multiple objects by providing a control-aware planning framework able to solve the problem of computing rest-to-rest transportation trajectories whose execution is less sensitive to parametric variations of the object dynamic properties. More specifically, the main innovation consisted of (i) a mathematical redefinition of the involved quantities to account for the joint presence of multiple objects and in (ii) a convenient derivation of a dynamically equivalent system representation that allows adopting the original optimal planning problem with minimal modifications involving only the number of non-sliding contact constraints.

The main novelty of the paper lies in the proposed dual-phase non-prehensile manipulation framework, which combines intentional sliding control and robust non-sliding transportation in a realistic tray-delivery scenario. In this framework, the LuGre–Limit Surface model and the NMPC controller constitute the main contribution for the tray setting phase, whereas the closed-loop-sensitivity-based planner and its extension to multiple objects constitute the main contribution for the transportation phase. In particular, this article provides the following contributions:

- A dual-phase model-based framework for tray manipulation, combining controlled sliding for tray setting and robust non-sliding motion for tray transportation.
- A planar sliding model based on LuGre friction and an ellipsoidal Limit Surface.
- An NMPC-based controller that actively regulates the tray inclination to drive an object toward a desired position through intentional sliding.
- A closed-loop sensitivity-based trajectory planner extended to multiple-object tray transportation, computing robust rest-to-rest motions while robustly enforcing non-sliding constraints for all objects under uncertain dynamic parameters.
- An experimental validation on a dual-robot tray-delivery scenario, integrating object placement, controlled sliding, and robust transportation.

2. Materials and Methods

In the following sections, we present the methodology we adopted for both the tray setting and its delivery phase. Specifically, Section 2.1 details the analytical model we derived to mathematically describe the phenomenon of an object sliding on a flat surface. Section 2.2 details the control design adopted to move the robot for adjusting the tray angles and letting the object slide to reach the desired position. Finally, Section 2.4 describes the algorithms developed to avoid slippage of one or more objects during tray transportation with uncertain parameters. Note that the solutions proposed in Sections 2.1 and 2.4 adopt two different analytical models. Indeed, while the first is formulated in SE(2), and thus describes only planar motions, the second extends the model to SE(3), thus allowing for the description of full three-dimensional rigid-body motions. This choice stems from the need to reduce the computational time required to solve the optimization problem formulated for the controlled sliding task (Section 2.2), which is performed online and must meet hard real-time constraints. However, the simplified model does not represent a limitation within our framework, as long as we assume that the accelerations involved are sufficiently low to allow the object to slide without losing contact with the tray. Conversely, the transportation phase might involve higher accelerations, and thus modelling the full motion is essential; however, the planning problem is solved offline, and no hard real-time constraint is required in this phase.

2.1. Dynamic Model of the Slider

Referring to Figure 2, we derive the dynamical model of an object sliding on a plane using the classical equations of rigid body mechanics ([43], Chapter 8). Let $\mathcal{F}_T$ be an inertial frame placed at the center of the tray, and $\mathcal{F}_S$ be the body frame, centered at its center of mass, which we assume is coinciding with the object's geometric center. We model the slider as a planar rigid body, thus characterized by two translational degrees of freedom $p_S = \begin{bmatrix} x_S & y_S \end{bmatrix}^{\mathrm{T}}$, and one rotational, θ_S.

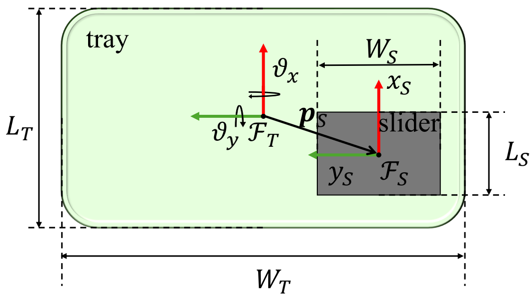


Figure 2. Mechanical system modelling for a planar sliding maneuver.

We denote the following:

- $M_S = \mathrm{blockdiag}(mI_{2\times2}, I_z)$ the inertia matrix, being $I_{2\times2}$ the two-dimensional identity matrix, m the slider mass, and I_z its moment of inertia along the z axis;
- $v_S = R_T^S(\theta_S)\dot{p}_S^T$, the slider linear velocity in the tray plane with respect to $\mathcal{F}_T$ expressed in the body frame ($R_T^S(\theta_S)$ is the 2-dimensional rotation matrix representing the orientation of $\mathcal{F}_T$ with respect to $\mathcal{F}_S$ –vectors with no superscript are expressed in the body frame $\mathcal{F}_S$);
- $\omega_S = \dot{\theta}_S$, the angular velocity of the slider;
- $\mathcal{V}_S = \begin{pmatrix} v_S^{\mathrm{T}} & \omega_S \end{pmatrix}^{\mathrm{T}}$, the 2-dimensional twist;
- $\dot{\mathcal{V}}_S$, the time derivative of $\mathcal{V}_S$ as defined above;
- $S(\cdot)$, the 2-dimensional skew-symmetric operator;
- $\tilde{S}(\omega_S) = \mathrm{blockdiag}(S(\omega_S), 0)$, the Coriolis matrix.

The equation of motion of the slider with respect to its center of mass can be written as

$$M_S\dot{\mathcal{V}}_S + \tilde{S}(\omega_S)M_S\mathcal{V}_S + \mathcal{W}_g = -\mathcal{W}_{TS}, \tag{1}$$

where $\mathcal{W}_{TS}$ represents the wrench exerted by the tray on the slider, and $\mathcal{W}_g = \mathcal{P}f_g$ is the gravitational wrench acting on the tray, given the gravitational force

$$f_g = R_x(\theta_x)R_y(\theta_y)ma_g, \qquad \mathcal{P} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \tag{2}$$

Specifically, being $a_g = \begin{pmatrix} 0 & 0 & -g \end{pmatrix}^{\mathrm{T}}$, where g is the gravity acceleration magnitude, the force f_g depends on the tray inclination angles θ_x and θ_y about the x and y directions, respectively, which define the rotation matrices

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\sin\theta_x \\ 0 & \sin\theta_x & \cos\theta_x \end{bmatrix}, \qquad R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \sin\theta_y \\ 0 & 1 & 0 \\ -\sin\theta_y & 0 & \cos\theta_y \end{bmatrix}. \tag{3}$$

Moreover, the wrench $\mathcal{W}_{TS}$ includes both the dry friction according to the LuGre model and the viscous one according to a linear velocity-proportional model. Inspired by [44] and following the ideas by [45] used to model the motion of a planar slider grasped by a parallel gripper equipped with soft tactile sensors, we adopt a roto-translational friction model based on both the classic LuGre dynamic model [38] and on the well-known ellipsoidal approximation of the Limit Surface [46]. In detail, the friction wrench exerted by the tray on the slider can be computed as

$$\begin{aligned} \mathcal{W}_{TS} &= \Sigma_0 z_{TS} + B_{TS}\mathcal{V}_S \\ \dot{z}_{TS} &= \mathcal{V}_S - \Sigma_0 \left(\mathcal{V}_S^{\mathrm{T}} L^{-1} \mathcal{V}_S\right)^{1/2} L z_{TS} \end{aligned}, \tag{4}$$

where

- $z_{TS} = \begin{pmatrix} z_t^{\mathrm{T}} & z_r \end{pmatrix}^{\mathrm{T}}$ is the internal state of the LuGre dynamic model, with z_t and z_r corresponding to the translational and rotational degrees of freedom, respectively;
- $\Sigma_0 = \mathrm{diag}(\sigma_{0_t}, \sigma_{0_t}, \sigma_{0_r})$ is the diagonal matrix containing the translational and rotational stiffness values of the LuGre model bristles;
- $L = \mathrm{diag}(1/f_{\max}^2, 1/f_{\max}^2, 1/\tau_{\max}^2)$, is the diagonal matrix describing the ellipsoidal Limit Surface, where
  - $f_{\max} = mg\mu_{TS}cos(\theta_x)cos(\theta_y)$ is the maximum linear friction force that the tray can apply to the slider, being μ_{TS} the static friction coefficient;
  - $\tau_{\max}$ is the maximum torsional friction moment that the tray can apply to the slider, and, assuming a uniform pressure distribution and denoting the area of the contact surface A with $a(A)$, can be computed as [47]

$$\tau_{\max} = \frac{mg\mu_{TS}}{a(A)} \int_A \|p^*\| dA\,; \tag{5}$$

- $B_{TS} = \mathrm{diag}(\beta_{t_{TS}}, \beta_{t_{TS}}, \beta_{r_{TS}})$ is the diagonal matrix of the translational ($\beta_{t_{TS}}$) and rotational ($\beta_{r_{TS}}$) viscous friction coefficients.

2.2. Nonlinear Model Predictive Control for Object Positioning

We exploited the nonlinear dynamic model described in (1) to set up a Nonlinear Model Predictive Controller (NMPC) and let the object slide on the tray to reach a target position $\bar{p}_S^T$ by following a path with an assigned time law $p_S^{T*}(t)$. In this regard, let $x = \begin{bmatrix} p_S^{T\mathrm{T}} & \theta_S & \mathcal{V}_S^{T\mathrm{T}} & z_{TS}^{\mathrm{T}} & \theta_x & \theta_y \end{bmatrix}^{\mathrm{T}}$ be the state vector, and $u = \begin{bmatrix} \dot{\theta}_x & \dot{\theta}_y \end{bmatrix}^{\mathrm{T}}$ be the control input, i.e., the angular rates of the tray along the x and y axis, respectively. Given the continuous-time nonlinear system dynamics in (1), by exploiting standard differential kinematics relationships, the continuous-time dynamics can be obtained in the form $\dot{x}(t) = f(x(t), u(t))$; we compute the corresponding discrete-time dynamics, i.e., $x_{k+1} = f_k(x_k, u_k)$, by employing an explicit step-adaptive solver with sample time T_s.

Based on the discrete-time model and given the prediction horizon H_p, and the control horizon $H_u \leq H_p$, we formulate the following finite-horizon optimal control problem (OCP) at time k:

$$\begin{aligned} \underset{u_0,\dots,u_{H_u}}{\text{minimize}} \quad & J = J_x + J_{\Delta_u} + J_{\text{out}} + J_\theta \\ \text{subject to} \quad & u_k \in \mathcal{U} \end{aligned}, \tag{6}$$

where $\mathcal{U} = [\dot{\theta}_{\min}, \dot{\theta}_{\max}] \times [\dot{\theta}_{\min}, \dot{\theta}_{\max}]$. Note that the system dynamics is moved outside the optimization problem, thanks to the recursive elimination discretization technique used to translate the optimal control problem in (6) to the standard formulation expected by the nonlinear programming algorithms [48]. Note that the control input sequence is optimized over the control horizon through a set of nodes selected within the interval $[0, H_p]$. This allows extending the control time window without increasing the control horizon H_u, and thus the dimensionality and the computational complexity of the optimization problem. Between consecutive nodes, we assume the control input to be constant using a zero-order hold (ZOH) scheme.

Let $\tilde{x}_k = x_k^* - x_k$ be the tracking error at time step k, and W_x and $W_{x_{H_p}}$ be the state weight matrices of proper dimensions, J_x in (6) is

$$J_x = \sum_{k=0}^{H_p-1} \tilde{x}_k^{\mathrm{T}} W_x \tilde{x}_k + \tilde{x}_{H_p}^{\mathrm{T}} W_{x_{H_p}} \tilde{x}_{H_p}. \tag{7}$$

Then, with $W_{\Delta u}$ being the control rate weight matrix, and $\Delta u_k = u_k - u_{k-1}$ the control rate, we compute $J_{\Delta u}$ as follows:

$$J_{\Delta u} = \sum_{k=1}^{H_p-1} \Delta {u_k}^{\mathrm{T}} W_{\Delta u} \Delta u_k. \tag{8}$$

To prevent the slider from falling off the tray, we introduce the additional state penalty J_{out} defined as a soft constraint in the cost function, i.e.,

$$J_{\text{out}} = \sum_{k=0}^{H_p-1} F_{\text{out}}(k) \quad \text{with } F_{\text{out}} = \begin{cases} p & \text{if } \left|x_S^T\right| \geq \dfrac{L_T}{2} - \dfrac{L_S}{2} \quad \text{or} \quad \left|y_S^T\right| \geq \dfrac{W_T}{2} - \dfrac{W_S}{2} \\ 0 & \text{otherwise,} \end{cases} \tag{9}$$

with p being the penalty factor, L_T, L_S, and W_T, W_S the tray and the object lengths along the x and y directions, respectively.

We also include the following soft penalty term J_θ in the cost function to discourage the tilt angles of the tray from exceeding a safety bound $\bar{\theta}_T$,

$$J_\theta = \sum_{k=0}^{H_p-1} f_p(\bar{\theta}_T - |\theta_{x_k}|) + f_p(\bar{\theta}_T - |\theta_{y_k}|), \tag{10}$$

where $f_p(\Delta\theta) = \alpha\, \sigma^2(\Delta\theta, \epsilon_\sigma)$, being $\sigma(\Delta\theta, \epsilon_\sigma) = \frac{1}{1+e^{\Delta\theta\epsilon_\sigma}}$, the sigmoid function.

Finally, according to the classic receding horizon approach, at each sampling time $t_k = kT_s$, we measure the state of the system, solve the OCP in (6), and apply the first element of the resulting optimal control trajectory to the system.

Despite our primary goal being the object positioning, we generate a reference trajectory $p_S^{T*}(t)$, resulting from a linear interpolation between the initial object position $p_{S_0}^T$ and the desired one, such that $p_S^{T*}(T_f) = \bar{p}_S^T$, where T_f denotes the duration of the initial ramp. The interpolated reference provides a progressively achievable setpoint, thus enhancing the convergence of the nonlinear solver with short prediction horizon. Note that we do not assign an angular reference trajectory, since it is not controllable by acting on the tray orientation only, thus the corresponding weight in W_x is simply set to zero. Moreover, we compute a reference trajectory for the tray angles to drive the tray back to its initial zero-angle (horizontal) position once the desired threshold ϵ_{track} is reached on the slider position errors $\tilde{p}_{Sx_k}^T$ and $\tilde{p}_{Sy_k}^T$ along the x and y directions, respectively. Thus, without altering the state weights, which could lead to instability of the system, we update the reference of the tray angles as follows:

$$\begin{cases} \theta_{x_k}^* & \text{if } |\tilde{p}_{Sx_k}^T| > \epsilon_{track} \\ 0 & \text{otherwise} \end{cases}, \qquad \begin{cases} \theta_{y_k}^* & \text{if } |\tilde{p}_{Sy_k}^T| > \epsilon_{track} \\ 0 & \text{otherwise} \end{cases}, \tag{11}$$

where $\theta_{x_k}^*$ and $\theta_{y_k}^*$ denote the k-th sample of a discrete-time linear interpolation between the current state and the desired final zero-degree tray configuration about the x and y directions, respectively, with $k = 0, \ldots, k_{end}$, and k_{end} being the total number of trajectory samples.

2.3. PD-like Controller for Object Positioning

In order to provide a comparison with baseline techniques which could be adopted for the sliding task, we also design a PD-like regulator. The nonlinear system in (1) is linearized around the equilibrium point $\theta_x = 0$, $\theta_y = 0$, and by considering the static nonlinear friction forces as disturbances. A linear controller for each direction is then designed by standard linear techniques in the Laplace domain, leading to the following PD-like transfer function,

$$\frac{k_c(s+z_1)(s+z_2)}{(s+p)^2}, \tag{12}$$

which is eventually discretized via the Tustin method at the same sampling time as the NMPC, i.e., 40 ms. Due to the hard nonlinear nature of the static friction, this controller alone was not capable of achieving a satisfactory performance; therefore, a nonlinear saturation action was necessary to allow the controller to limit the tilting angles to a value θ_T slightly above the minimum angle causing the object to slide under the gravitational force. In detail, the saturation action is designed as follows for the x direction

$$\dot{\theta}_x = \begin{cases} 0 & \text{if } \theta_x > \theta_T \;\wedge\; \dot{\vartheta}_x > 0 \\ 0 & \text{if } \theta_x < -\theta_T \;\wedge\; \dot{\vartheta}_x < 0; \\ \dot{\vartheta}_x & \text{otherwise} \end{cases} \tag{13}$$

where $\dot{\vartheta}_x$ represents the regulator output with no saturation applied. The definition for the y direction is analogous. Moreover, we saturate the control inputs such that it always results in $u_k \in [\dot{\theta}_m, \dot{\theta}_M] \times [\dot{\theta}_m, \dot{\theta}_M]$, and, as done for the NMPC, as soon as the object reaches the desired position with the set threshold ϵ_{track}, we allow the tray to recover the configuration with $\theta_x = \theta_y = 0$. In this case, we simply activate two proportional controllers ($k_p = 1$) with the objective to regulate the tray angles to zero.

2.4. Uncertainty-Aware Trajectory Planner for Tray Transportation

In this section, we describe the uncertainty-aware trajectory planning method adopted in this work to minimize the occurrence of object slippage during transportation in the face of parametric uncertainties of the object dynamics. Parametric uncertainties are handled at trajectory planning level leveraging the concept of closed-loop state sensitivity and the associated ellipsoidal tubes of perturbed trajectories [37]. This quantity captures how deviations in the model parameters (with respect to their nominal values) affect the closed-loop evolution of the robot/environment states. Our goal is to find a reference trajectory for the closed-loop system that, when tracked, exhibits minimum state sensitivity to system parametric uncertainties, thus increasing the intrinsic robustness of tracking. In our case, we considered as uncertain the mass (and consequently the moment of inertia) of the two objects transported on the tray. However, the framework we use is general and can easily handle other uncertain parameters, relevant for our case study, such as tray friction coefficient or object center of mass. In Section 2.4.1 we review the concept of closed-loop sensitivity, which is the main pillar behind our optimization framework, while in Section 2.4.2 we detail the typical optimization problem adopted for the scope. Our novel contribution is then presented in Sections 2.4.3 and 2.4.4 which extend the previous methods by proposing a model (and a controller) that account for the presence of multiple objects. From an operational viewpoint, we use the equations shown in Section 2.4.1, particularize them for the multi-object system introduced in Section 2.4.3 that is controlled via the control law presented in Section 2.4.4, and construct the optimization problem shown in Section 2.4.2 that offline calculates the coefficient of the optimal polynomial reference trajectory.

https://doi.org/10.3390/machines1010000

2.4.1. Closed-Loop Sensitivity

To minimize the possibility of object sliding, we exploit the notion of *closed-loop sensitivity*, which addresses the problem of robustness against a controlled system's parametric uncertainties. In our case, uncertainties may be about the manipulated object's mass, inertia, or its friction parameter, while the controller knows its nominal values.

From a mathematical viewpoint, generic system dynamics can be described by the following ordinary differential equation:

$$\dot{x} = f(x, u, p), \tag{14}$$

where $x \in \mathbb{R}^{n_x}$ represents the system state, $u \in \mathbb{R}^{n_u}$ is the control input, and $p \in \mathbb{R}^{n_p}$ is the vector of system parameters which might be uncertain. In the context of tracking tasks, let us consider an output function $y(x) \in \mathbb{R}^{n_y}$ (e.g., the position of the manipulated object) and its desired reference trajectory $y_d(t) \in \mathbb{R}^{n_y}$. A generic trajectory tracking controller can be written as

$$\begin{cases} \dot{\xi} = g(\xi, x, y_d(t), p_n, t), \\ u = h(\xi, x, y_d(t), p_n, t). \end{cases} \tag{15}$$

where $\xi \in \mathbb{R}^{n_\xi}$ denotes the internal state of the controller (e.g., in case of an integral action), p_n are the nominal values of p (the controller has no knowledge of the true system's parameters). For the closed-loop system it is possible to define the state sensitivity matrix and the input sensitivity matrix as

$$\Pi(t) = \frac{\partial x(t)}{\partial p}\Big|_{p=p_n} \in \mathbb{R}^{n_x \times n_p}, \qquad \Theta(t) = \frac{\partial u(t)}{\partial p}\Big|_{p=p_n} \in \mathbb{R}^{n_u \times n_p}, \tag{16}$$

which accounts for the variation of the state/input, respectively, caused by a variation in the parameters. In general, a closed-form expression for $\Pi(t)$ and $\Theta(t)$ is not available but it can be computed by forward integration of the following quantities:

$$\begin{cases} \dot{\Pi}(t) = \frac{\partial f}{\partial x}\Pi + \frac{\partial f}{\partial u}\Theta + \frac{\partial f}{\partial p}, \quad \Pi(t_0) = 0, \\ \dot{\Pi}_\xi(t) = \frac{\partial g}{\partial x}\Pi + \frac{\partial g}{\partial \xi}\Pi_\xi, \quad \Pi_\xi(t_0) = 0, \\ \dot{\Theta}(t) = \frac{\partial h}{\partial x}\Pi + \frac{\partial h}{\partial \xi}\Pi_\xi. \end{cases}$$

Matrices Π and Θ can be used to, e.g., generate the reference trajectory $y_d(t)$ that produces minimally sensitive behaviors [33], or to evaluate the tubes (or ellipsoids) enveloping uncertain states/inputs [37]. Assuming that each parameter p_i can vary in a given range of δp_i, centered at a nominal p_{c_i}, i.e.,

$$p_i \in [p_{c_i} - \delta p_i,\ p_{c_i} + \delta p_i], \tag{17}$$

and letting $\Delta p = p - p_c$, an ellipsoid in the parameter space centered at p_c and with semi-axes δp_i has the equation

$$\Delta p^{\mathrm{T}} W^{-1} \Delta p = 1,$$

with $W = \mathrm{diag}(\delta p_i^2)$ being a diagonal weight matrix. Exploiting (16) enables the derivation of the corresponding ellipsoids in the state space, by defining $\Delta x = x - x_c$, i.e.,

$$\Delta x^{\mathrm{T}} (\Pi W \Pi^{\mathrm{T}})^{-1} \Delta x = 1, \qquad \Delta u^{\mathrm{T}} (\Theta W \Theta^{\mathrm{T}})^{-1} \Delta u = 1, \tag{18}$$

https://doi.org/10.3390/machines1010000

where x_c is the state evolution of (14) and (15) in the unperturbed case ($p = p_c$), and in the input space with analogous definition of $\Delta u = u - u_c$.

The state and input space ellipsoids can be used for different purposes. In particular, in the context of this work, one can exploit (18) for obtaining the tubes of perturbed trajectories for the individual components of the states/inputs. In the case of inputs, for each direction of interest n_i in the input space, one can obtain the tube radius r_i as

$$r_i = \sqrt{n_i^{\mathrm{T}} \Theta W \Theta^{\mathrm{T}} n_i}. \tag{19}$$

This radius can then be exploited to build an envelope for the input as follows:

$$u_{c,i} - r_i \leq u_i \leq u_{c,i} + r_i, \tag{20}$$

with $u_{c,i}$ being the behavior of the i-th input in the unperturbed case ($p = p_c$). Equation (20) bounds all the perturbed inputs when the parameter uncertainty is bounded as in (17). The same approach can be applied to compute the envelope of the perturbed states x, or of any quantity whose sensitivity matrix is available.

2.4.2. Optimization Problem

Thanks to the construction of tubes and state-of-the-art approaches [34,37], we propose the use of sensitivity to avoid actuator saturation, exploiting Θ, or to avoid collision with the environment, leveraging Π. In this work, the sensitivity tubes are utilized to ensure feasibility of the control law. Indeed, as shown in [17], based on the system's model, it is possible to build an optimal trajectory guaranteeing that the object will not slide. It takes the form $y^*(t) = \left(q^{o*}, \mathcal{V}^{o*}, \dot{\mathcal{V}}^{o*}\right)$, where $q^{o*} \in \mathrm{SE}(3)$ denotes the object configuration in the inertial frame, with $\mathcal{V}^{o*} \in \mathbb{R}^6$ representing its twist and $\dot{\mathcal{V}}^{o*} \in \mathbb{R}^6$ representing its acceleration in the object frame. However, in the presence of parameter uncertainty, e.g., the object mass, tracking this optimal trajectory may still cause the sliding of the mass.

To avoid this, it is possible to employ the trajectory optimization problem in order to exploit the aforementioned closed-loop sensitivity concepts. Firstly, the desired trajectory can be expressed as an n-th order polynomial with a set of coefficients $a \in \mathbb{R}^{n \times 6}$ such that $y^*(t) = y^*(a, t)$. Starting from this, the goal is to compute the optimal curve parameters a which minimize some function of the sensitivity matrix Π (for instance its two-norm) along the executed trajectory. Formally,

$$a = \arg\min \int_0^{T_f} ||\Pi(t)||^2 \mathrm{d}\tau \tag{21}$$

$$\text{s.t.} \quad \dot{x} = f(x, u, p) \tag{22}$$

$$y^*(a, t_0) = y_0, \quad y^*(a, t_f) = y_f \tag{23}$$

$$\Lambda(t) - r_\Lambda(t) > 0 \qquad \forall t \in [t_0, t_f] \tag{24}$$

where T_f represents the trajectory time duration, $y_0 = \left(q_0^o, \mathcal{V}_0^o, \dot{\mathcal{V}}_0^o\right)$ and $y_f = \left(q_f^o, \mathcal{V}_f^o, \dot{\mathcal{V}}_f^o\right)$ represent the initial and final desired trajectory conditions, respectively. As better explained in the next section, the last inequality enforces the sought non-sliding condition, with $\Lambda(t)$ and $r_\Lambda(t)$ being the vector of coefficients of the contact forces and of their corresponding tube radii, respectively, computed via (19).

The first and second constraints guarantee that the optimal trajectory respect the system dynamics and is compliant with the boundary conditions requested by the specific task. The third constraint, instead, represents the non-sliding constraint and it is the key integration of the closed-loop sensitivity theory within the non-prehensile transportation problem at hand. Indeed, thanks to the definition of the tubes, it is possible to guarantee

https://doi.org/10.3390/machines1010000

that even in the worst-case scenario, i.e., the lowest perturbed input, the contact forces stay within the friction cone, minimizing the possibility to slide.

In summary, the goal of the optimization problem in (21) is twofold. It aims at building an optimal trajectory $y^*(a, t)$ such that: (i) the norm of the state sensitivity matrix along the path is minimized, i.e., the system behavior will be as robust as possible; (ii) the control action is always feasible, even in the presence of parameter uncertainties.

2.4.3. Extension to Multiple Objects

In this section, we extend the previous version of the trajectory optimizer, originally presented in [21], accounting for the joint presence of a number $n_o \in \mathbb{Z}_{>1}$ cuboid objects on the tray.

We describe the model of the multiple-objects system with reference to Figure 3 where a generic system comprising multiple objects is represented. We denote with $q_i = (p_i, R_i) \in$ SE(3) with $i \in \{1, \ldots, n_o\}$, the current pose of the $i-$th object frame $\mathcal{O}_i$, fixed to the object center of mass, in the tray reference frame, where $p_i \in \mathbb{R}^3$ denotes the $i-$th object position vector, and $R_i \in$ SO(3) its orientation. We associate to each body a set of $n_c = 4$ contact points located along the object perimeter of the object's face in contact with the tray. The pose of $j-$th contact point in $\mathcal{O}_i$ is known and identified by a contact frame $\mathcal{C}_j^{o_i} \in$ SE(3). We assume all the contacts behave according to the point contacts with friction model [49], i.e., only linear forces $f_{c_j}^{o_i} \in \mathbb{R}^3$, can be transmitted through the i-th contact. This allows expressing the wrench $\mathcal{W}^{o_i} = (f^{o_i}, \tau^{o_i}) \in \mathbb{R}^6$, with $f^{o_i}, \tau^{o_i} \in \mathbb{R}^3$ force and torque vectors, respectively, as

$$\mathcal{W}^{o_i} = G^{o_i} F_c^{o_i}, \quad \text{with} \quad G^{o_i} = \left[\mathrm{Ad}^{\mathrm{T}}_{q^{-1}_{o_i,c_1}} B_c, \ldots, \mathrm{Ad}^{\mathrm{T}}_{q^{-1}_{o_i,c_{n_c}}} B_c \right], \qquad \forall i = 1, \ldots, n_o, \tag{25}$$

where $G^{o_i} \in \mathbb{R}^{6 \times 3n_c}$ maps the stacked vector of independent contact forces $F_c^{o,i} = [f_{c_1}^{o_i\mathrm{T}}, \ldots, f_{c_{n_c}}^{o_i\mathrm{T}}]^{\mathrm{T}} \in \mathbb{R}^{3n_c}$ to the body wrench $\mathcal{W}^{o,i}$ exerted at the object center of mass, the symbol $\mathrm{Ad}^{\mathrm{T}}_{q^{-1}_{o_i,c_j}}$ expresses the adjoint matrix corresponding to the transformation $q^{-1}_{o_i,c_j}$, while the matrix $B_c \in \mathbb{R}^{6 \times 3}$ accounts for the previously introduced contact model and maps the components of the contact force that can be transmitted across the contact point, into the 6-dimensional space. With these assumptions, matrices involved in the calculation of G can be written as follows:

$$\mathrm{Ad}^{\mathrm{T}}_{q^{-1}_{o_i,c_j}} = \begin{bmatrix} R_{o_i,c_j} & 0 \\ -R_{o_i,c_j}\hat{p}_{c_j,o_i} & R_{o_i,c_i} \end{bmatrix}, \quad \text{and} \quad B_c = \begin{bmatrix} I_{3\times 3} \\ 0_{3\times 3} \end{bmatrix}, \quad \forall i = 1, \ldots, n_o, \quad \forall j = 1, \ldots, n_c. \tag{26}$$

where $\hat{p}_{o_i,c_j} \in$ so(3) denotes the skew-symmetric matrix associated with the vector $p_{o_i,c_j} \in \mathbb{R}^3$, i.e., the position of the j-th contact point expressed in $\mathcal{O}_i$.

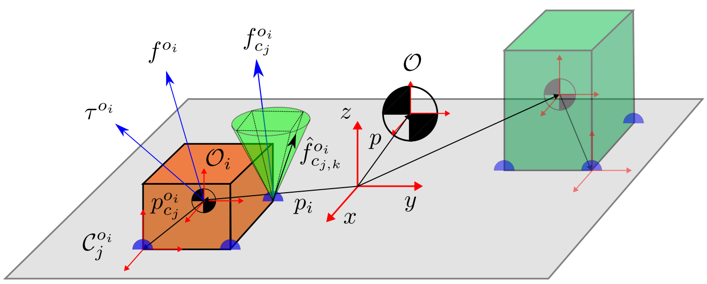


Figure 3. Illustration of the multiple-objects tray-based non-prehensile manipulation model. Each object is subject to the external wrench $\mathcal{W}^{o_i} = (f^{o_i}, \tau^{o_i})$ produced by frictional forces $f_{c_j}^{o_i}$ exerted by the tray.

The set of generalized contact forces that can be physically realized belong to the friction cone. In this work, we use a linearly approximated representation of the composite friction cone space $FC = \cdots \times FC^{o_i}_{c_j} \times \cdots \subset \mathbb{R}^{3n_o \times n_c}$, where each $FC^{o_i}_{c_j}$ is conservatively approximated with a cone-inscribed polyhedral generated by m vectors $\hat{f}_{c_j}$. In this case, choosing $m = 4$ the circular cone is approximated by a pyramid as shown in Figure 3. This is a common and conservative approximation which allows us to treat the (otherwise second order) friction cone as a linear constraint in the following optimization problem. Indeed, the condition $f^{o_i}_{c_j} \in FC^{o_i}_j$ can be conveniently formulated expressing $f^{o_i}_{c_j}$ as a nonnegative linear combination of unit vectors $\hat{f}^{o_i}_{c_j,1}, \ldots, \hat{f}^{o_i}_{c_j,m}$ i.e.,

$$FC^{o_i}_{c_j} = \left\{ f^{o_i}_{c_j} \in \mathbb{R}^3 \;\middle|\; f^{o_i}_{c_j} = \sum_{k=1}^{m} \lambda^{o_i}_{c_j,k} \hat{f}^{o_i}_{c_j,k},\ \lambda^{o_i}_{c_j,k} \geq 0 \right\}, \qquad \forall i = 1, \ldots, n_o, \quad \forall j = 1, \ldots, n_c. \tag{27}$$

The set of feasible contact forces are those belonging to the linearized friction cone space in (27) and can be compactly written as

$$F^{o_i}_c = \hat{F}^{o_i}_c \Lambda^{o_i}, \quad \text{with} \quad \Lambda^{o_i} = [\lambda^{o_i}_{1,1}, \ldots, \lambda^{o_i}_{n_c,m}]^{\mathrm{T}} \geq 0, \qquad \forall i = 1, \ldots, n_o, \tag{28}$$

where the last inequality is intended component-wise and

$$\hat{F}^{o_i}_c = \mathrm{diag}\left(\hat{F}^{o_i}_{c,1}, \ldots, \hat{F}^{o_i}_{c,n_c}\right), \quad \hat{F}^{o_i}_{c,i} = \left[\hat{f}^{o_i}_{c,1}, \ldots, \hat{f}^{o_i}_{c,k}\right], \qquad \forall i = 1, \ldots, n_o.$$

In summary, substituting $F^{o_i}_c$ from (28) in (25) yields

$$\mathcal{W}^{o_i} = G^{o_i} \hat{F}^{o_i}_c \Lambda^{o_i}, \quad \text{with} \quad \Lambda^{o_i} \geq 0, \qquad \forall i = 1, \ldots, n_o. \tag{29}$$

Iterating this reasoning for all the objects, the non-sliding condition for the multi-objects system is expressed by enforcing non-negativeness of all the components of the vector $\Lambda^o = [\Lambda^{o_1 \mathrm{T}}, \ldots, \Lambda^{o_n \mathrm{T}}]^{\mathrm{T}}$. Indeed, when all the components are positive, friction forces are entirely inside the friction cone space defined earlier, making the objects stationary to the tray. As some components approach zero, the corresponding contacts might be sliding. To face the considered dynamic uncertainties in the model, the non-sliding condition is enforced in (21) via tubes introduced in Section 2.4.1.

In view of the enforced non-sliding behavior, we can assume the multiple-object system to be rigid (no relative motion can occur between the objects). This allows us to derive its mathematical, dynamically equivalent, representation, computing the system dynamics with total mass M, center of gravity p coincident with the origin of the frame $\mathcal{O}$, and mass moment of inertia I about the common reference frame, that match those of the initial system. For n_o objects, this translates into the following quantities

$$M = \sum_{i=1}^{n_o} m_i, \qquad p = \frac{\sum_{i=1}^{n_o} m_i p_i}{\sum_{i=1}^{n_o} m_i}, \qquad I = \sum_{i=1}^{n_o} I_i - m_i S^2(p_i) \tag{30}$$

where m_i represent the i-th object mass, p_i the i-th object center of mass position, I_i the i-th object center of mass inertia matrix, and $S(\cdot)$ the three-dimensional skew-symmetric operator. To enforce non-sliding behavior of the multiple objects on the tray, we formulate the tray multi-object system dynamics using the parameters computed in (30) and use the relative non-sliding constraints in the optimization problem (21) to ensure that contact forces belong to the corresponding friction cones even in the presence of uncertainties. To this end, all the original system contact points (4 considered for each object) are used to formulate these constraints.

The multi-object system shown in Figure 3 is described by the dynamic model [50]

$$M^o \dot{\mathcal{V}}^o + C^o(\mathcal{V}^o)\mathcal{V}^o + N^o(R_o) = \mathcal{W}^o, \tag{31}$$

where $M^o \in \mathbb{R}^{6\times 6}$ is the constant, positive-definite mass matrix built from the total object mass $M > 0$ and inertia matrix $I \in \mathbb{R}^{3\times 3}$ expressed by (30), C^o captures centrifugal and Coriolis effects, and N^o accounts for gravity. The twist $\mathcal{V}^o = (v^o, \omega^o) \in \mathbb{R}^6$ contains the linear and angular velocities. The total wrench $\mathcal{W}^o = (f^o, \tau^o) \in \mathbb{R}^6$ on the right hand side of (31) can be expressed in view of (29) as follows

$$\mathcal{W}^o = \bar{G}\hat{F}_c^o \Lambda^o, \qquad \Lambda \geq 0 \tag{32}$$

where $\bar{G} = \text{blockdiag}(\text{Ad}^{\text{T}}_{q^{-1}_{o,o_1}} G^{o_1}, \dots, \text{Ad}^{\text{T}}_{q^{-1}_{o,o_{n_o}}} G^{o_{n_o}})$, and $\hat{F}_c^o = \text{blockdiag}(\hat{F}_c^{o_1}, \dots, \hat{F}_c^{o_{n_o}})$.

2.4.4. Multi-Object System Controller

The consistency of the trajectory optimization problem in (21) is ensured by the inclusion of the model of the closed-loop system dynamics. The object model in (31), computed for the dynamically equivalent multi-object system exploiting (30), describes how the system evolves when subject to external wrench which is produced by frictional forces. Assuming the external wrench $\mathcal{W}^o$ is expressed as in (29), a suitable control law to track a desired trajectory $y^*(t) = \left(q^{o*}, \mathcal{V}^{o*}, \dot{\mathcal{V}}^{o*}\right)$ can be devised as follows

$$\begin{aligned}\Lambda = \left(\bar{G}\hat{F}_c^o\right)^\dagger \Big(& C^o(\mathcal{V}^o)\mathcal{V}^o + N^o(R^o) + M^o\big(\dot{\mathcal{V}}^{o*} + \\ & K_d\left(\mathcal{V}^{o*}(t) - \mathcal{V}^o\right) + K_p(q^{o*} - q^o)\big)\Big).\end{aligned} \tag{33}$$

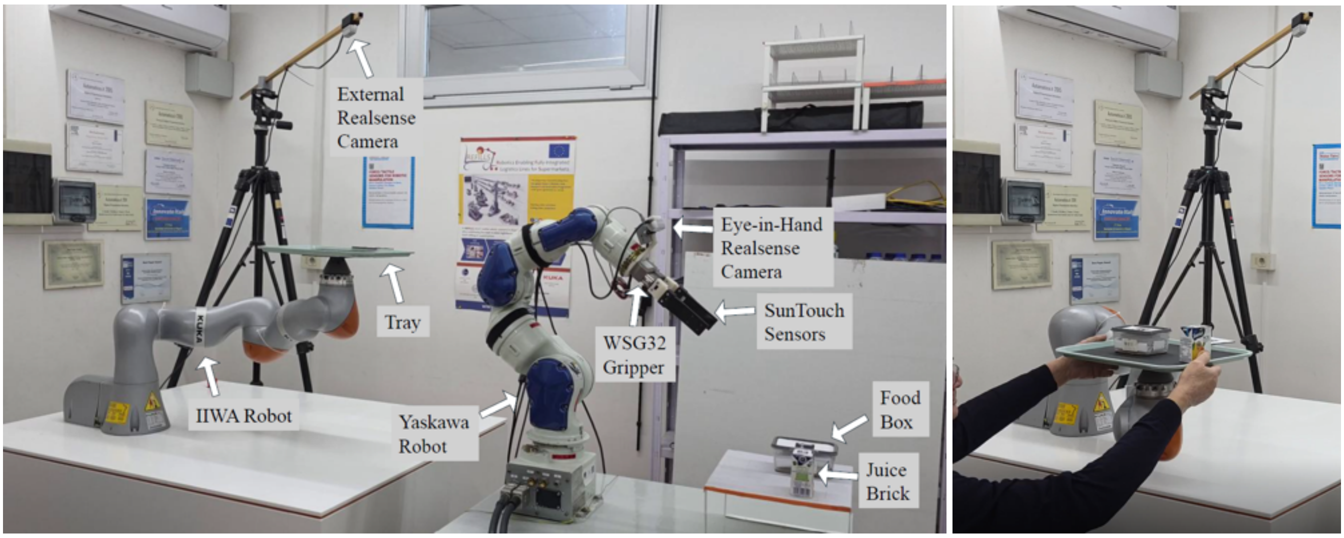
Substituting (33) in (32) and this in (31), it is trivial to show that the proposed control achieves the asymptotic tracking of the desired time varying trajectory imposing the following system dynamics

$$\dot{\mathcal{V}}^{o*} - \dot{\mathcal{V}}^o + K_d(\mathcal{V}^{o*} - \mathcal{V}^o) + K_p(q^{o*} - q^o) = 0.$$

Note that the use of the Moore–Penrose pseudoinverse of $G\hat{F}_c$ in (33) (represented via the symbol †) does not guarantee that $\Lambda \geq 0$ (which corresponds to contact forces feasibility) but results in the minimization of the 2-norm of Λ when the constraint $\Lambda \geq 0$ is not binding. However, the constraint relative to Λ imposed in (21) ensures that the resulting optimal trajectory allows the robust feasibility of the contact forces. In this way, the possibility of objects sliding is avoided.

3. Results

Figure 4 shows the experimental setup adopted for the real-world controlled sliding experiments, as well as the final case study proposed in Section 3.3. It consists of placing the food box and the juice brick on the tray, then delivered through the tray transportation task to a patient (Figure 4b). Specifically, the IIWA robot holds the tray and performs the controlled sliding maneuver by relying on an external RealSense camera to track the objects at 30 Hz, while the Yaskawa SIA5F robot is responsible for placing the objects on the tray. Thus, it is equipped with a WSG32 gripper, SunTouch tactile sensors [51], and an Intel RealSense camera in an eye-in-hand configuration.



(**a**) Dual-arm setup (**b**) Task Goal

Figure 4. Experimental setup.

3.1. Controlled Object Sliding

In order to measure the state of the system for the tray setting task, we exploit the robot's forward kinematics, which allows us to retrieve θ_x and θ_y, and two ArUco markers, placed on the sliding object and on the tray (see Figure 5), which allow us to track the slider planar position p_S^T and its orientation θ_S. Then, we compute the slider planar twist $\mathcal{V}_S^T$ via finite differences. Finally, we cannot directly measure the LuGre state variables, thus we simply initialize them to zeros at each prediction step. Note that this choice has a negligible effect on the controller performance, thanks to the fast dynamics of the LuGre friction model.

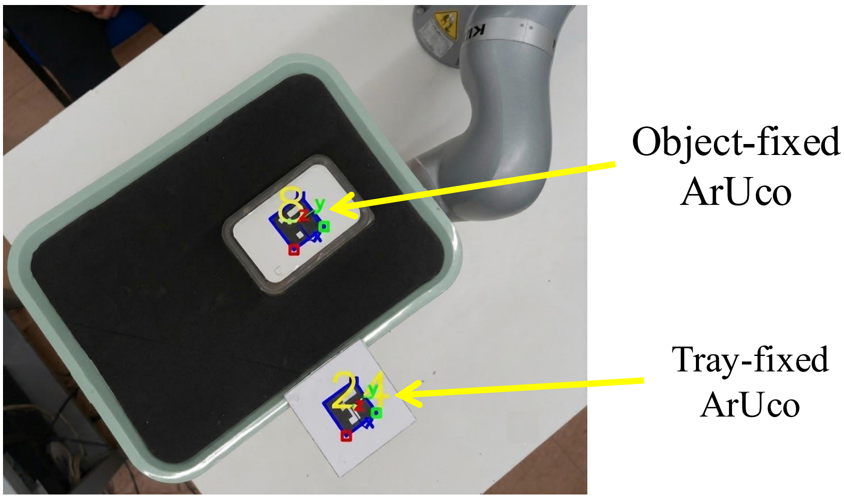


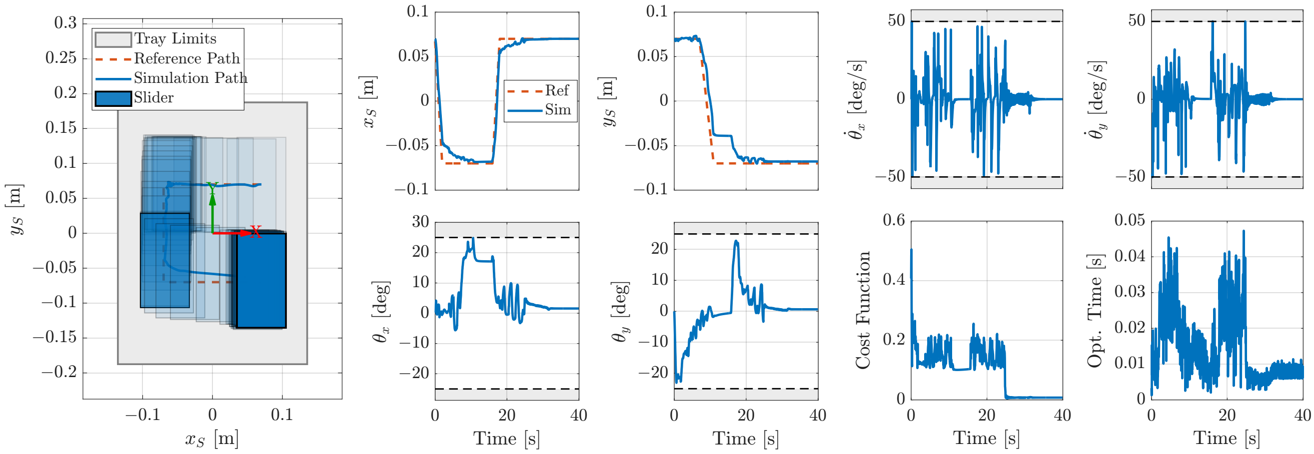
Figure 5. ArUco setup for object tracking.

Given the NMPC parameters in Table 1a and the objects' parameters in Table 1c, experimentally identified according to procedures similar to those described in [45], we first validated our NMPC and the PD-like controller by designing two different use cases performed in simulation only. The first one, hereafter referred to as Case Study A, consists of letting the slider move along the edges of the tray, therefore the corresponding trajectory is defined by three different waypoints; in the second one, referred to as Case Study B, the robot has to let the object slide from the top-left to the bottom-right corner. Note that, despite the reference trajectory does not simply consists of a final position to be reached, as explained in Section 2.2, our primary objective is object positioning rather than trajectory tracking. Therefore, to compare the performance of the two proposed controllers, we evaluate both the norm of the steady-state positioning error and the norm of the maximum tracking error along x and y, as well as their settling time, computed starting from the instant T_f, at which the reference trajectory reaches its final constant value. Consequently, the ramp duration is excluded from the settling time calculation. Despite the deterministic nature of the optimization algorithm used by the NMPC, and considering the difficulty in placing the objects in the same initial position and the variability of the friction properties of the tray surface, as well as the noise affecting the measurements that we reproduced in simulation, we repeated the task of each case study five times and reported the three indicators in terms of their mean (and standard deviation).

Table 1. Control parameters of the proposed controllers and physical parameters of the object-tray system.

Parameter	Value
a. NMPC Parameters for Controlled Sliding	
Parameter	Value
H_u (samples)	$\{0, 3, 6\}$
H_p	10
W_x	$\mathrm{diag}(1, 1, 0_{1\times 7}, 0.1, 0.1)$
$W_{x_{H_p}}$	$2000 W_x$
$W_{\Delta \boldsymbol{u}}$	$[0.005, 0.005]$
$\dot{\theta}_{\min}$	-50 deg/s
$\dot{\theta}_{\max}$	50 deg/s
p	10,000
$\bar{\theta}_T$	25 deg
α	500
ϵ_σ	100
ϵ_{track}	0.007 m
T_s	0.04 s
b. PD-like controller parameters	
Parameter	Value
k_c	66
z_1	1.5
z_2	8
p	20
θ_T	12 deg
k_p	1
$\dot{\theta}_m$	-30 deg/s
$\dot{\theta}_M$	30 deg/s
c. Food box and tray parameters	
Parameter	Value
m	0.265 kg
W_S	0.137 m
W_T	0.375 m
L_S	0.091 m
L_T	0.271 m
μ_{TS}	0.346
$f_{\max}$	0.909 N
$\tau_{\max}$	0.0247 Nm
σ_{0_t}	10,000 N/m
σ_{0_r}	1 Nm/rad
$\beta_{t_{TS}}$	2.15 Ns/m
$\beta_{r_{TS}}$	0.0045 Nms/rad

Figures 6 and 7 show the results of Case Study A achieved by the NMPC and the PD-like controller, respectively. As the metrics reported in Table 2 also highlight, the NMPC outperforms the PD-like controller. Specifically, it achieves lower steady-state positioning and tracking error, as well as a reduced settling time, thus highlighting its superior control performance and faster convergence.



(**a**) XY-plane task visualization (**b**) Tracking performance (**c**) Control input and solver metrics

Figure 6. Case Study A: NMPC simulation results.

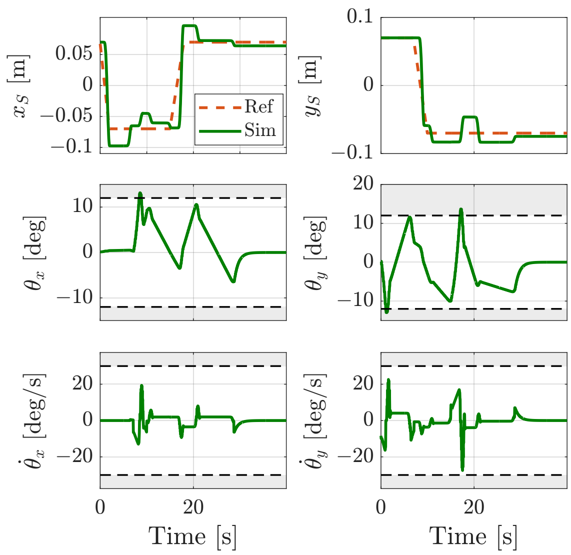
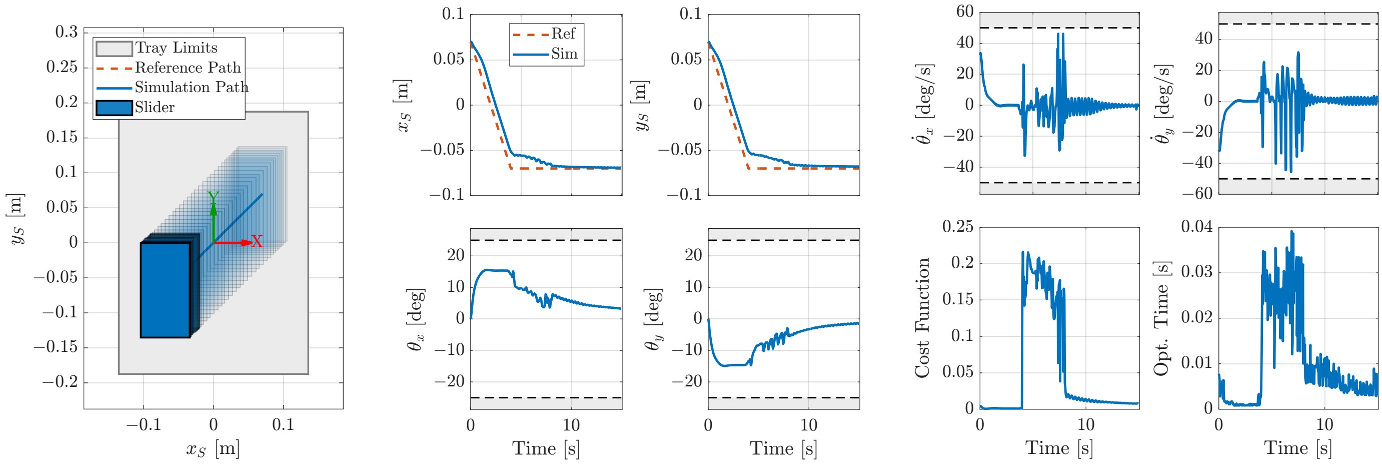


Figure 7. Case Study A: PD-like simulation results.

Table 2. Case Study A: mean(standard deviation) of the performance metrics.

Waypoint and Controller	Steady-State Error [m]	Maximum Tracking Error [m]	Settling Time [s]
Waypoint 1 (PD-like)	0.0048 (3.1×10^{-4})	0.071 (3.3×10^{-4})	4.0 (0.30)
Waypoint 1 (NMPC)	0.0069 (1.7×10^{-4})	0.028 (2.1×10^{-4})	3.8 (0.20)
Waypoint 2 (PD-like)	0.0085 (2.4×10^{-4})	0.064 (4.3×10^{-4})	10.2 (1.2)
Waypoint 2 (NMPC)	0.014 (1.1×10^{-4})	0.039 (2.2×10^{-4})	8.0 (0.15)
Waypoint 3 (PD-like)	0.0066 (3.5×10^{-4})	0.063 (2.7×10^{-4})	5.1 (0.72)
Waypoint 3 (NMPC)	0.0021 (1.3×10^{-4})	0.026 (1.7×10^{-4})	1.1 (0.17)

Figures 8 and 9 show the results of Case Study B for the NMPC, and the PD-like controller, respectively. Again, the metrics reported in Table 3 confirm the superiority of the proposed NMPC over the baseline PD-like controller even in this case study. Indeed, both steady-state and maximum tracking errors are significantly reduced by using the nonlinear model-predictive controller, which also allows for a minor reduction of the settling time.



(**a**) XY-plane task visualization (**b**) Tracking performance (**c**) Control input and solver metrics

Figure 8. Case Study B: NMPC simulation results.

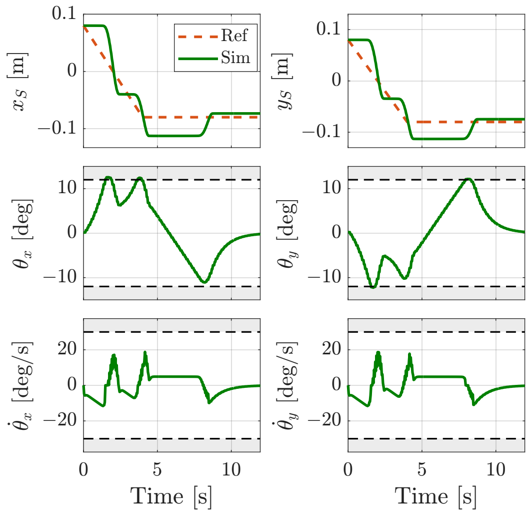


Figure 9. Case Study B: PD-like simulation results.

Table 3. Case Study B: mean(standard deviation) of the performance metrics.

Controllers	Steady-State Error [m]	Maximum Tracking Error [m]	Settling Time [s]
PD-like	0.0085 (2.1×10^{-4})	0.047 (2.2×10^{-4})	4.8 (0.4)
NMPC	0.0024 (1.7×10^{-4})	0.0276 (2.1×10^{-4})	4.0 (0.2)

The proposed case studies clearly demonstrate the superior performance of the NMPC compared to the PD-like controller. However, the NMPC highly rely on the system dynamic model to predict the state trajectory along the prediction horizon and compute the optimal input trajectory. As a result, its performance may be sensitive to uncertainties affecting the physical parameters involved in the proposed dynamic model. Therefore, we also provide a robustness analysis of the designed NMPC to evaluate the effect of variations of the slider mass m, and the friction coefficient between the slider and the tray μ_{TS}. Note that we decided to carry out the analysis in simulation only, so that we can rely on a controlled environment where known and verifiable uncertainties can be easily introduced. In this regard, we intentionally modified the aforementioned parameters only in the model used by the predictor within the predictive controller. Specifically, we varied each parameter independently by $\pm 50\%$ and $\pm 25\%$ with respect to the nominal values presented in

Table 1c. We then evaluated the mean of the steady-state positioning error along the x, and y directions, while asking the controller to execute the sliding maneuver of Figure 8. Figure 10 shows that the designed controller is robust against uncertainties affecting m and μ_{TS} parameters. The presented analysis highlights that, under uncertainty, the controller robustness is enhanced by overestimating the friction coefficient, or equivalently underestimating the object mass, rather than the vice versa scenario. This result is reasonable within the predictive control framework; indeed, when the friction coefficient is overestimated, the predictor model assumes that a larger tray inclination is required to initiate the object sliding. Therefore, the controller computes more aggressive control actions, which result in higher tray inclination angles to compensate for the overestimated dry friction. Despite this modelling uncertainty, the high measurement update frequency allows the controller to rapidly adjust the control inputs, thereby achieving overall good performance in the tracking task. Conversely, when the friction coefficient is underestimated, the model predicts object sliding even with small inclination angles. The controller, therefore, applies control actions that, according to its predictions, would allow the slider to move but are too small to overcome the real dry friction. The mismatch between predicted and actual system behavior leads, in this scenario, the controller to anticipate the sliding motion, resulting in overshoots and chattering in the control inputs, which lead to a higher RMSE.

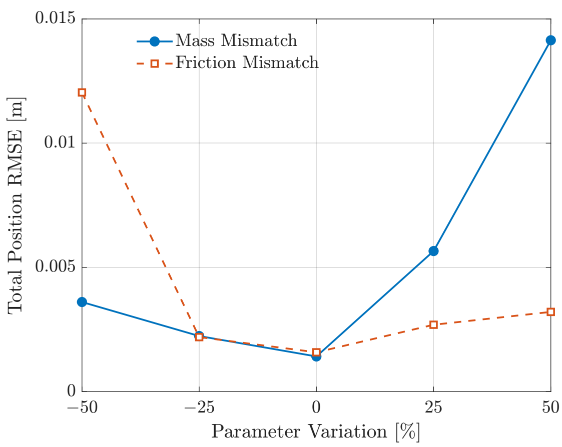
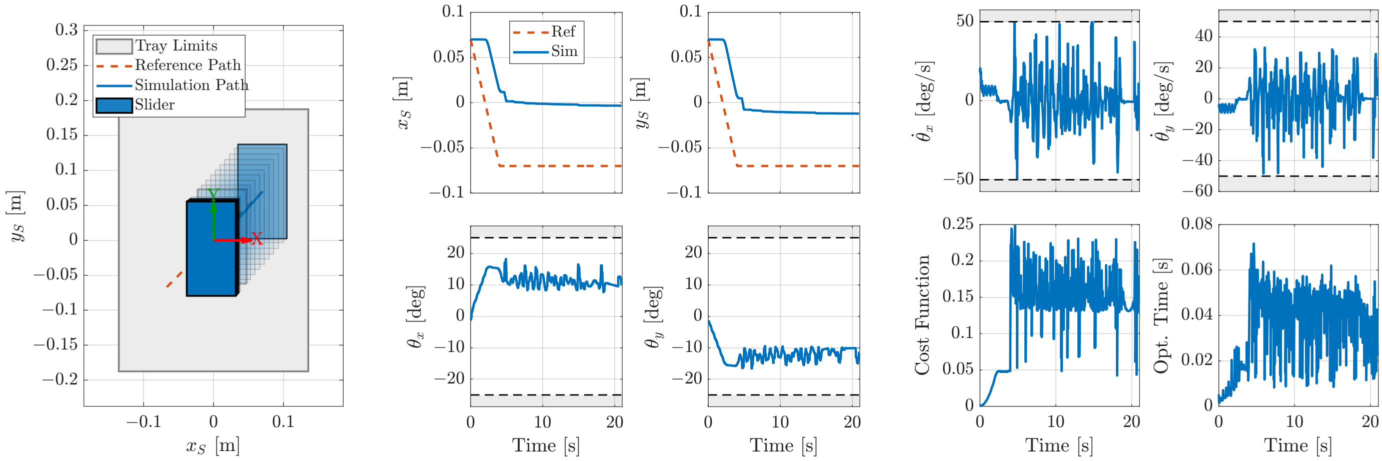


Figure 10. NMPC robustness analysis.

To further assess the importance of modeling the friction effects during the sliding maneuvers into the prediction model used by the NMPC, we also conduct an ablation analysis, which consists of neglecting the controller's knowledge of dry friction. Therefore, the friction contributions to the dynamic model in (1), arising from the LuGre state variables, is simply set to zero in the prediction model of the controller, whose parameters are still those presented in Table 1a. Figure 11 clearly demonstrates the necessity of including friction effects in the prediction model to achieve accurate tracking and more stable behavior. Indeed, the lack of friction knowledge make the NMPC unable to perform the task, as it lead the controller to predict an excessive sliding of the object over the tray surface. As a result, the optimizer struggles to find the optimal control trajectory, as also the computational time of the optimization steps in Figure 11c shows.

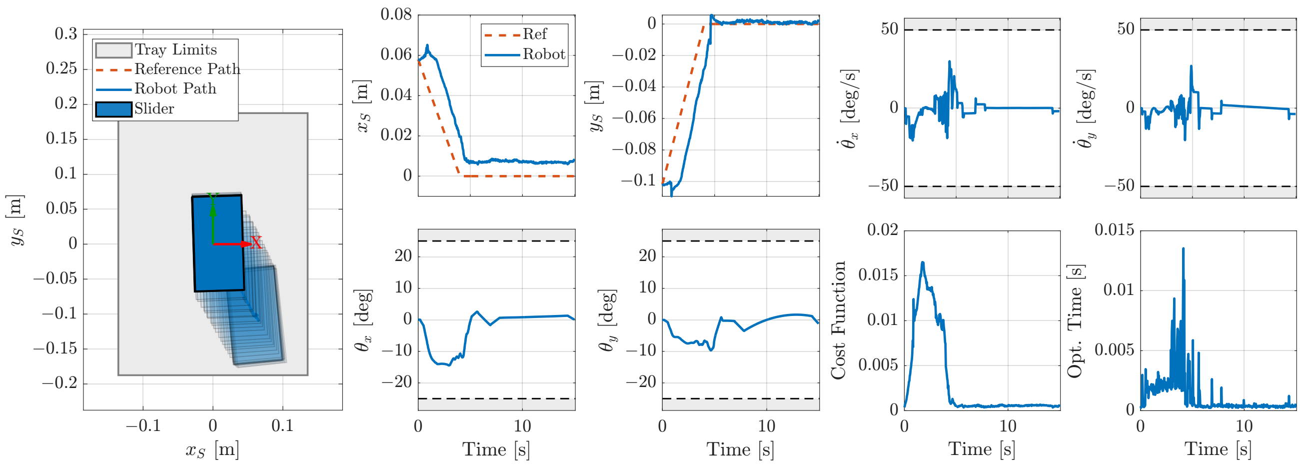
Finally, we designed two further case studies to experimentally validate both the controllers in a real-world scenario. The first one, referred to as Case Study C, consists of letting the object slide from the top-right corner to the center of the tray; the second, referred to as Case Study D, consists of letting the object slide from the tray center to its right edge.



(**a**) XY-plane task visualization (**b**) Tracking performance (**c**) Control input and solver metrics

Figure 11. Ablation analysis results of Case Study B.

Figures 12 and 13, and Table 4 show the results achieved by the NMPC and the PD-like controller and the corresponding performance metrics for Case Study C, respectively, whereas Figures 14 and 15, and Table 5 illustrate the results and the performance metrics corresponding to the Case Study D, respectively. The results of both Case Study C and D confirm the outcomes already observed in the simulation. Indeed, despite the uncertainties that unavoidably affect the physical parameters of the prediction model, the NMPC outperforms the PD-like controller in terms of steady-state error, maximum tracking error and settling time.



(**a**) XY-plane task visualization (**b**) Tracking performance (**c**) Control input and solver metrics

Figure 12. Case Study C: NMPC experimental results.

Table 4. Case Study C: mean(standard deviation) of the performance metrics.

Controllers	Steady-State Error [m]	Maximum Tracking Error [m]	Settling Time [s]
PD-like	0.0059 (3.1×10^{-4})	0.0625 (2.7×10^{-4})	31.3 (1.1)
NMPC	0.0051 (2.4×10^{-4})	0.049 (1.9×10^{-4})	1.5 (0.1)

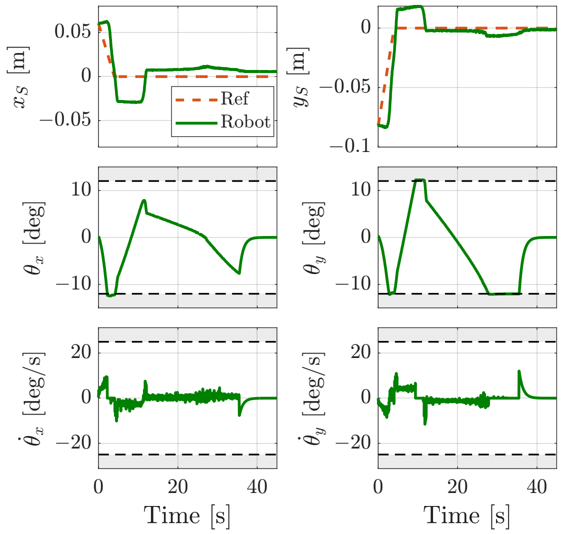
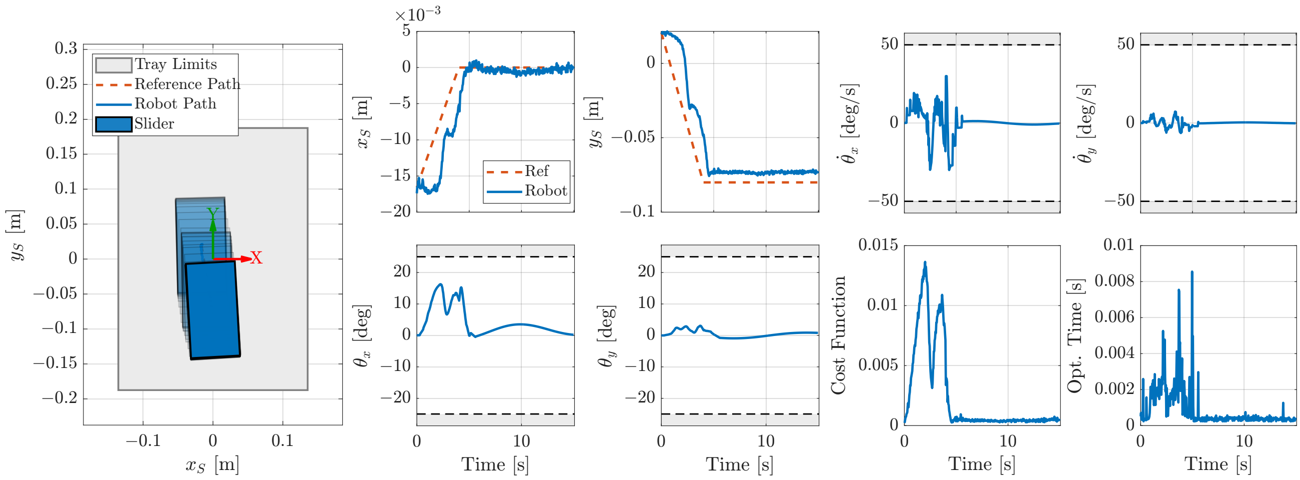


Figure 13. Case Study C: PD-like experimental results.



(**a**) XY-plane task visualization (**b**) Tracking performance (**c**) Control input and solver metrics

Figure 14. Case Study D: NMPC experimental results.

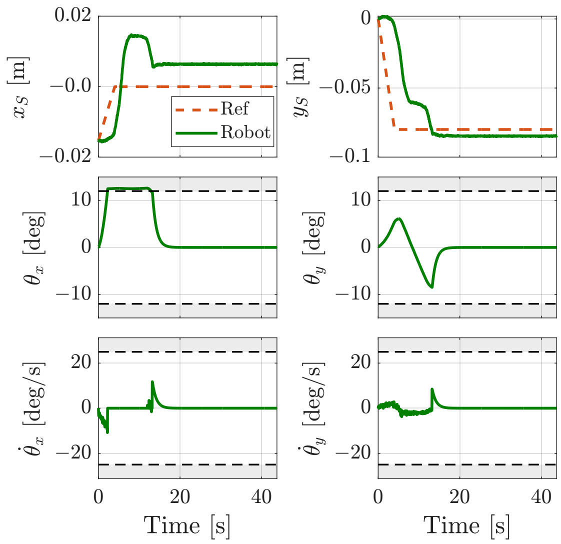


Figure 15. Case Study D: PD-like experimental results.

Table 5. Case Study D: mean(standard deviation) of the performance metrics.

Controllers	Steady-State Error [m]	Maximum Tracking Error [m]	Settling Time [s]
PD-like	0.0079 (2.6×10^{-4})	0.076 (3.8×10^{-4})	10 (0.81)
NMPC	0.0075 (2.3×10^{-4})	0.0428 (3.4×10^{-4})	0.60 (0.24)

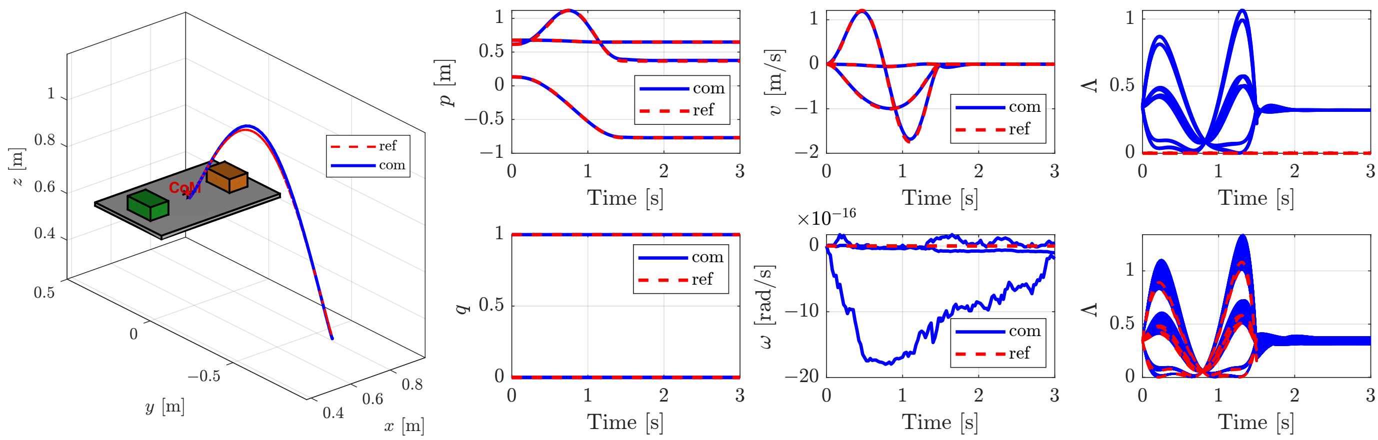
3.2. Tray Transportation

Here we describe the details of the implementation and the simulation of the robust trajectory planning method introduced in Section 2.4 to move the tray set with all objects from an initial pose to a target one without any slip of the objects. We consider a setup with a 0.6×0.4 m tray carrying two cuboid objects with dimensions $0.06 \times 0.04 \times 0.02$ m whose centers of mass are placed at $(0.2, 0.0, 0.1)$ m and $(-0.2, 0.0, 0.1)$ m with respect to the tray center. The mass of the two objects is 0.265 kg and their principal moment of inertia are $(2.25, 2.25, 2.25) \times 10^{-3}$ kg m^2. Using these data in (30) leads to

$$M = 0.53 \text{ kg} \qquad p = (0, 0, 0.1) \text{ m} \qquad I = \text{diag}(9.8, 31.0, 25.7) \times 10^{-3} \text{ kg m}^2. \tag{34}$$

which are used in the dynamic model (31). We assume uniform friction with coefficient $\mu = 0.35$. The first dynamic constraint equation in (21) is constructed making use of (33) where $K_d = \text{diag}(10, 10, 10, 0.2, 0.2, 0.2)$ and $K_p = \text{diag}(70, 70, 70, 2, 2, 2)$, and the system dynamics in (31). The second constraint, accounting for trajectory initial and final point have been selected imposing a rest-to-rest trajectory between $p_i = (0.677, 0.132, 0.615)$ m and $p_f = (0.649, -0.770, 0.367)$ m and upright orientation coincident with the one shown in Figure 3. The vector a, denoting coefficients of a degree 8 polynomial trajectory, encode this constraints in the linear non-homogeneous form $Ma = d$, where M and d are the linear coefficient matrix and the vector of constants, respectively. The solution corresponding to a straight line between initial and end configurations is used as initial guess for the optimization problem. Trajectory duration has been selected as $T_f = 1.5$ s. The third constraint is built making use of (33). The constrained minimization problem in (21) has been formulated and implemented using CASADI in MATLAB R2025b and considers an uncertainty on the object masses (and consequently on their moment of inertia) of 10%. The solution of the optimization problem was computed offline and required 16 s computation time with a code non-optimized for speed.

Results are reported in Figure 16. On the left, in Figure 16a, we show the tray setup and the optimal desired trajectory represented as a red dashed line (ref) computed from the solution a^*. We also plot in blue the actual trajectory followed by the system center of mass (com) while tracking the optimal one. Tracking performance can be appreciated on the right hand side, in Figure 16b, where we plot, in the first two columns, position p, quaternion q, linear velocity v, and angular velocity ω desired (ref) and actual (com) trajectories. Finally, the last column on the right reports in blue the time evolution of the inputs Λ for the two objects in case of nominal parameters (top), and in case of perturbed parameters (bottom) along 100 simulations performed considering random variability of the objects masses within the 10% range. In the former case, we can see that Λ components never fall below their zero lower bound shown in red dashed line, thus respecting the imposed non-sliding condition. In the latter case, we report, using red dashed lines, the lower bound of the tubes encapsulating Λ trajectories in the perturbed case, capturing variability on the inputs Λ, yet never violating non-sliding conditions. Coherently with previous works [21], the resulting optimal trajectory is predominantly characterized by a bell shape along the z axis. This trajectory minimizes the sensitivity of the state to parameter variation along the execution of the task, thus minimizing the possibility of object sliding in the presence of uncertainty.



(**a**) Task visualization (**b**) Tracking performance

Figure 16. Simulation result: (**a**) visualization of the optimal (ref) and actual (com) trajectories for the tray with two objects system; (**b**) trajectory tracking performance and control inputs respecting the non-sliding condition.

3.3. Final Case Study

By exploiting the ArUco marker and the tactile fingertips, the Yaskawa robot detects and grasps the food box (see Figure 17a). Due to the workspace limitations of this robot, the only available placement position on the tray is the one depicted in Figure 17b, at the right edge of the tray. Given this configuration, a controlled sliding maneuver is required to move the food box already on the tray, and thus make the placement of the juice brick feasible. Referring to Figure 17c, the IIWA robot performs the non-prehensile maneuver, letting the food box slide on the tray to reach the desired position.

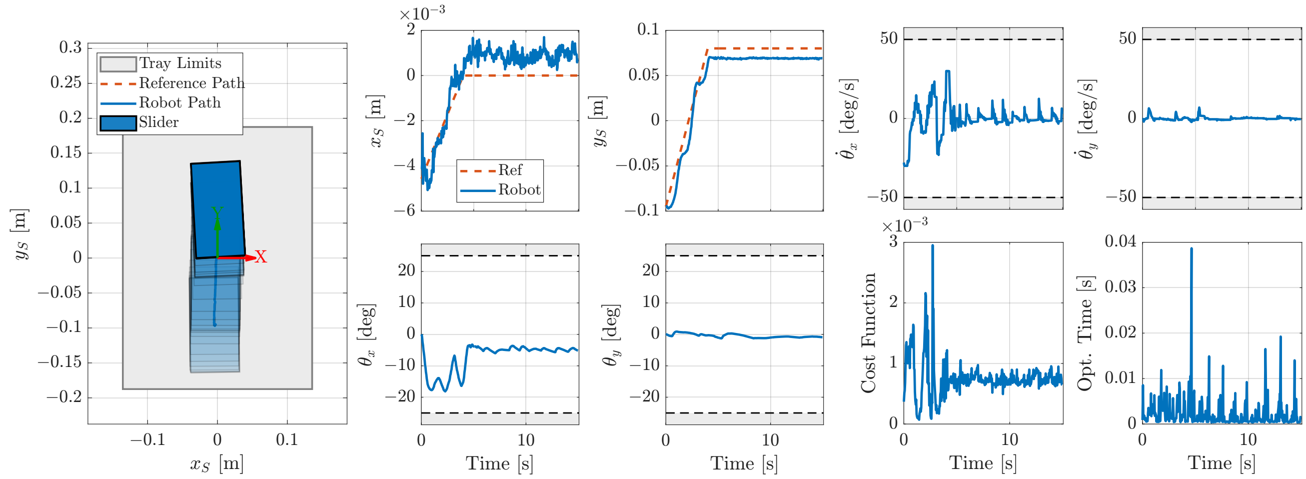
Figure 18 shows the results of the tray setting task. The NMPC is able to bring the food box to the desired position, reaching an error within the defined threshold ϵ_{track}.

(**a**) Food box picking (**b**) Food box placing (**c**) Controlled food box sliding

(**d**) Juice brick picking (**e**) Juice brick placing (**f**) Tray Delivering

Figure 17. Tray setting and delivering pipeline.

https://doi.org/10.3390/machines1010000

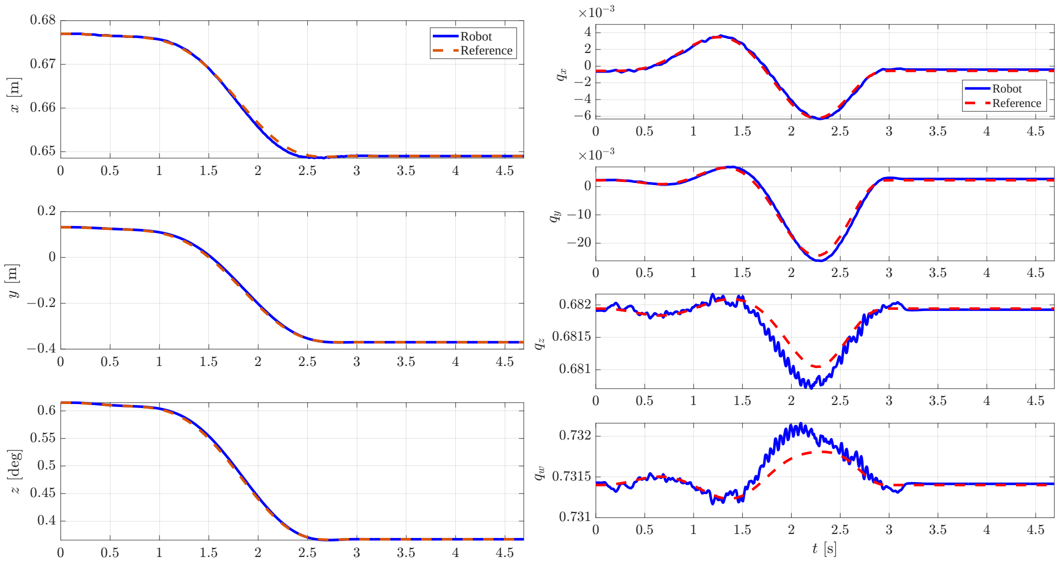


(**a**) XY-plane task visualization (**b**) Tracking performance (**c**) Control input and solver metrics

Figure 18. Experimental results of right edge to tray center setting task.

In the meantime, the Yaskawa robot detects the juice brick by adopting the DOPE convolutional neural network [52] combined with a pose refinement algorithm [53], and grasps it (Figure 17d). Note that the ArUco marker on this object is not used for its detection, but only to quantitatively evaluate the quality of the delivery at the end of the experiment. Thanks to the object re-placement, the Yaskawa robot can easily place the juice brick on the tray (Figure 17e), thus concluding the tray setting phase of the overall experiment.

Finally, once the tray is ready for delivery, the IIWA robot performs the tray transportation and approaches the patient workspace (Figure 17f). Specifically, the planning algorithm described in Section 2.4 generates a safe 6D tray trajectory that causes a negligible object sliding on the tray (see Figure 19).



(**a**) Position tracking along x, y, and z directions (**b**) Quaternion tracking

Figure 19. Trajectory tracking during the tray transportation phase.

The transportation result can be measured in terms of the amount of object slippage; to this aim, two photos are taken before and after the maneuver. By comparing the ArUco markers with the one on the tray, it is possible to estimate the total slippage, given the intrinsic parameters of the camera. At the end of the experiment, the xy slippage of the objects was a maximum of 3%, and 1% of the tray dimensions for the juice brick and the food box, respectively.

The Supplementary Material provides a video of the overall final case study execution.

4. Discussion

The final case study presented in Section 3.3 highlights the effectiveness of the proposed framework, showcasing the advantages of employing non-prehensile manipulation actions in a real-world scenario where robots with limited workspace are adopted for an assistive-like task. The predictive controller designed to perform the sliding tasks demonstrated high tracking accuracy, as well as robustness against parameter uncertainties. Moreover, to satisfy real-time constraints, we had to select a 40 ms sampling time. However, state-of-the-art research has shown that a learning-based approach could help to address the computational issue and allow for lower sampling times. Indeed, several works [54,55] have proved the efficiency of Physics-Informed Neural Network (PINN) in a predictive model-based framework. Specifically, a physics-informed simulation model can be trained by exploiting the highly accurate mathematical model and can be used as an efficient-to-evaluate surrogate model within the MPC. Indeed, trained surrogate models are considerably faster to evaluate since they do not rely on numerical integration algorithms, which often represent a bottleneck for speeding up the solution of optimization problems. On the other hand, any learning-based approach requires sufficient training data and computational power for training the model.

The proposed uncertainty-aware trajectory planning method allows robust tray-based multi-object transportation via a dynamically equivalent object model. A dynamically equivalent system model usually simplifies a more complex (or even distributed-mass) system into a streamlined framework of one or more concentrated point masses. It is a well- established principle in mechanics and is used for instance to find lumped-mass approximation of slider-crank mechanisms [56,57]. By satisfying the three core mathematical conditions in (30), the equivalent model maintains the exact same equations of motion, natural frequencies, inertial/potential energy, and responses to external loads as the original system model even in case of significantly different original masses. In other words, having masses that differ by several orders of magnitude does not, by itself, reduce the accuracy of the dynamically equivalent representation. It is worth remarking that, when building a dynamically equivalent system, we have not taken additional assumptions or approximation, so equivalence conditions mentioned above are exact and do not introduce additional any form of modeling errors. For this reason, we have not quantitatively evaluated the error between the single-body model and the actual multi-contact model. We believe that, if it exists, this error is not significant in our case study as our uncertainty-aware robust planning approach is shown to handle successfully the envisioned situation.

We leverage the closed-loop sensitivity concept considering uncertainty concentrated primarily in the object mass (and, consequently, in its inertia). In real-world service robot scenarios the greatest uncertainty is usually in the coefficient of friction. However, for what concerns our application scenario of non-sliding transport trajectory execution, we believe robustness with respect to static friction uncertainties can be handled in a simpler and more straightforward way by enforcing a more conservative approximation of the friction cone. Moreover, the focus of our work is on trajectory tracking performance which is not really affected by friction uncertainty (unless objects are sliding) rather than by mass parameter uncertainty (which is known to significantly affect any feedback linearizing controller). Another reason for ignoring friction uncertainty in this work, is the conservative position we take enforcing that every component of the Λ vector is greater that zero: under real-world working conditions we have noticed that some components can become negative without producing sliding. For all these reasons, we believe considering mass variation constitutes a more interesting research problem in our application scenario. It is worth remarking that the robust planning framework based on state sensitivity, however, is general and can handle any parametric uncertainty affecting the real system (such as friction, but also center of mass, which are quantities that are not directly measurable in static or non-prehensile configurations) with minimal variation to the code.

About the non-slip condition, enforced as optimization constraint, this is derived for the linearized friction cone. However, as the linear friction cone conservatively approximate the real one adhering to the Coulomb model (see e.g., [17]), guaranteed constraint satisfaction theoretically preserves the non-slip condition also with real-world Coulomb friction. Another conservative position we take is enforcing that every component of the Λ vector must be greater that zero: in real world conditions, some components can become negative without producing sliding.

As every novel method proposed to solve a technological problem, our approach has some limitations. One is related to the use of ArUco markers, which might be impractical in a real setting. If placing a QR-code-like marker on the lid of a food container can be realistic, placing it in a calibrated place of the tray to be transported might be more difficult. Therefore, the need for a reliable visual tracking algorithm is of paramount importance if it is to be used in a real-time feedback loop. However, existing tracking algorithms are not yet efficient and reliable enough to be used, e.g., in a real hospital setting with changing lighting conditions and a large variety of objects. Possible solutions might exploit homography-based algorithms [58] or recent learning-based techniques [59].

Another limitation regards the uncertainty-aware planning method based on closed-loop sensitivity that allows planning locally optimal trajectories that exhibit minimal variability under parametric uncertainties, starting from an initial guess, which, in our case is assumed a straight line trajectory. In future work we aim to investigate the combination of the proposed local planner, possibly with global ones, to avoid getting stuck in local minima. Further limitations include: the necessity of deriving the symbolic expression of the system model and its relative constraints, which require a non-negligible setup time of the optimization problem in case of model changes, such as in the number of objects, or in their relative configuration; as the number of objects increases, the number of constraints to be considered in the trajectory optimization increases, thus it directly affects the solution time of the optimization problem. However, as the trajectory planning optimization approach is designed to be executed offline, these limitations do not constitute a real problem in our application scenario, and, as long as the constraints are all satisfied, no object motion can occur.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/machines1010000/s1.

Author Contributions: Conceptualization, M.C. and M.S.; methodology, S.F., M.C. and M.S.; software, S.F. and M.S.; validation, S.F. and M.C.; writing—original draft preparation, C.N.; writing—review and editing, S.F., M.C., C.N., M.S. and F.R.; supervision, C.N.; funding acquisition, M.C. and F.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Italian Ministry of University and Research (MUR) under the European Union - Next-GenerationEU - Piano Nazionale di Ripresa e Resilienza (PNRR) grant number P2022MHR5C (DARC project) and under the Fondo Italiano per la Scienza (FIS) - FIS 3 (ROMOTIVE Project).

Data Availability Statement: Data is contained within the article or Supplementary Material.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bowen, J.T.; Morosan, C. Beware hospitality industry: The robots are coming. *Worldw. Hosp. Tour. Themes* **2018**, *10*, 726–733.
2. Basori, A.H. NAOCared: Intelligent and Communally Humanoid assistive robot for elderly care support. *Int. J. Comput. Sci. Netw. Secur.* **2020**, *20*, 113–120.
3. Bicchi, A. On the Closure Properties of Robotic Grasping. *Int. J. Robot. Res.* **1995**, *14*, 319–334. https://doi.org/10.1177/027836499501400402.
4. Ruggiero, F.; Lippiello, V.; Siciliano, B. Nonprehensile dynamic manipulation: A survey. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1711–1718.
5. Bertoncelli, F.; Ruggiero, F.; Sabattini, L. Linear Time-Varying MPC for Nonprehensile Object Manipulation with a Nonholonomic Mobile Robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Paris, France, 31 May–31 August 2020; pp. 11032–11038. https://doi.org/10.1109/ICRA40945.2020.9197173.
6. Satici, A.; Ruggiero, F.; Lippiello, V.; Siciliano, B. Coordinate-free framework for robotic pizza tossing and catching. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 3932–3939.
7. Serra, D.; Ruggiero, F.; Donaire, A.; Buonocore, L.; Lippiello, V.; Siciliano, B. Control of nonprehensile planar rolling manipulation: A passivity-based approach. *IEEE Trans. Robot.* **2019**, *35*, 317–329.
8. Raei, H.; De Momi, E.; Ajoudani, A. A Reinforcement Learning Approach to Non-Prehensile Manipulation Through Sliding. *IEEE Robot. Autom. Lett.* **2025**, *10*, 7715–7722. https://doi.org/10.1109/LRA.2025.3578238.
9. Erdmann, M.; Mason, M.T. An Exploration of Sensorless Manipulation. *Proc. IEEE J. Robot. Autom.* **1988**, *4*, 369–379.
10. Peshkin, M.A.; Sanderson, A.C. Manipulation of a sliding object. In Proceedings of the (ICRA) International Conference on Robotics and Automation, Francisco, CA, USA, 7–10 April 1986; pp. 233–239.
11. Vose, T.H.; Umbanhowar, P.; Lynch, K.M. Sliding Manipulation of Rigid Bodies on a Controlled 6-DoF Plate. In *Robotics: Science and Systems VII*; Durrant-Whyte, H., Roy, N., Abbeel, P., Eds.; The MIT Press: Cambridge, MA, USA, 2012. https://doi.org/10.7551/mitpress/9481.003.0048.
12. Das, A.; Bollimuntha, S.; Jeevesh, M.V.R.; Patra, K.; Ghosh, T.; Govindan, N.; Singh, A.K.; Krishna, K.M. DART: Learning-Enhanced Model Predictive Control for Dual-Arm Non-Prehensile Manipulation. *arXiv* **2026**, arXiv:2604.17833.
13. Selvaggio, M.; Cacace, J.; Pacchierotti, C.; Ruggiero, F.; Robuffo Giordano, P. A shared-control teleoperation architecture for nonprehensile object transportation. *IEEE Trans. Robot.* **2021**, *38*, 569–583.
14. Miguel Garcia-Haro, J.; Martinez, S.; Balaguer, C. Balance Computation of Objects Transported on a Tray by a Humanoid Robot Based on 3D Dynamic Slopes. In Proceedings of the 2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids), Beijing, China, 6–9 November 2018; pp. 1–6. https://doi.org/10.1109/HUMANOIDS.2018.8624920.
15. Martucci, G.; Bimbo, J.; Prattichizzo, D.; Malvezzi, M. Maintaining stable grasps during highly dynamic robot trajectories. In Proceedings of the 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021; pp. 9198–9204. https://doi.org/10.1109/IROS45743.2020.9341575.
16. Subburaman, R.; Selvaggio, M.; Ruggiero, F. A non-prehensile object transportation framework with adaptive tilting based on quadratic programming. *IEEE Robot. Autom. Lett.* **2023**, *8*, 3581–3588.
17. Selvaggio, M.; Garg, A.; Ruggiero, F.; Oriolo, G.; Siciliano, B. Non-Prehensile Object Transportation via Model Predictive Non-Sliding Manipulation Control. *IEEE Trans. Control Syst. Technol.* **2023**, *31*, 2231–2244. https://doi.org/10.1109/TCST.2023.3277224.
18. Heins, A.; Schoellig, A.P. Keep It Upright: Model Predictive Control for Nonprehensile Object Transportation with Obstacle Avoidance on a Mobile Manipulator. *IEEE Robot. Autom. Lett.* **2023**, *8*, 7986–7993. https://doi.org/10.1109/LRA.2023.3324520.
19. Morlando, V.; Selvaggio, M.; Ruggiero, F. Nonprehensile Object Transportation with a Legged Manipulator. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 6628–6634. https://doi.org/10.1109/ICRA46639.2022.9811810.
20. Yang, W.; Posa, M. Dynamic On-Palm Manipulation via Controlled Sliding. *arXiv* **2024**, arXiv:2405.08731.
21. Teimoorzadeh, A.; Pupa, A.; Selvaggio, M.; Haddadin, S. Robust Nonprehensile Dynamic Object Transportation: A Closed-Loop Sensitivity Approach. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025; pp. 9192–9198. https://doi.org/10.1109/ICRA55743.2025.11127885.
22. Astrom, K.J.; Wittenmark, B. *Adaptive Control*, 2nd ed.; Addison-Wesley: Boston, MA, USA, 1994.
23. Ortega, R.; Van Der Schaft, A.; Maschke, B.; Escobar, G. Interconnection and damping assignment passivity-based control of port-controlled Hamiltonian systems. *Automatica* **2002**, *38*, 585–596.
24. Teimoorzadeh, A.; Donaire, A.; Arpenti, P.; Ruggiero, F. Robust energy shaping for mechanical systems with dissipative forces and disturbances. In Proceedings of the 2022 European Control Conference (ECC), London, UK, 12–15 July 2022; pp. 1409–1414.
25. Agboh, W.C.; Dogar, M.R. Pushing Fast and Slow: Task-Adaptive Planning for Non-prehensile Manipulation Under Uncertainty. In *Proceedings of the Algorithmic Foundations of Robotics XIII*; Morales, M., Tapia, L., Sánchez-Ante, G., Hutchinson, S., Eds., Springer International Publishing: Cham, Switzerland, 2020; pp. 160–176.

26. Jawale, N.; Boots, B.; Sundaralingam, B.; Bhardwaj, M. Dynamic Non-Prehensile Object Transport via Model-Predictive Reinforcement Learning. In Proceedings of the 2025 IEEE International Conference on Robotics and Automation (ICRA), Atlanta, GA, USA, 19–23 May 2025; pp. 3647–3653. https://doi.org/10.1109/ICRA55743.2025.11127521.
27. Hwang, J.; Yang, T.; Jeong, J.; Yoon, M.; Yoon, S.E. Uncertainty-Aware Non-Prehensile Manipulation with Mobile Manipulators under Object-Induced Occlusion. In Proceedings of the 2026 IEEE International Conference on Robotics and Automation (ICRA), Vienna, Austria, 1–5 June 2026; pp. 14321–14328.
28. Candido, S.; Hutchinson, S. Minimum uncertainty robot path planning using a POMDP approach. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; pp. 1408–1413.
29. Candido, S.; Hutchinson, S. Minimum uncertainty robot navigation using information-guided POMDP planning. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 6102–6108.
30. Ansari, A.; Murphey, T. Minimum sensitivity control for planning with parametric and hybrid uncertainty. *Int. J. Robot. Res.* **2016**, *35*, 823–839.
31. Tedrake, R.; Manchester, I.R.; Tobenkin, M.; Roberts, J.W. LQR-trees: Feedback motion planning via sums-of-squares verification. *Int. J. Robot. Res.* **2010**, *29*, 1038–1052.
32. Majumdar, A.; Tedrake, R. Funnel libraries for real-time robust feedback motion planning. *Int. J. Robot. Res.* **2017**, *36*, 947–982.
33. Robuffo Giordano, P.; Delamare, Q.; Franchi, A. Trajectory generation for minimum closed-loop state sensitivity. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 286–293.
34. Srour, A.; Franchi, A.; Robuffo Giordano, P. Controller and trajectory optimization for a quadrotor uav with parametric uncertainty. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 1–7.
35. Pupa, A.; Robuffo Giordano, P.; Secchi, C. Optimal energy tank initialization for minimum sensitivity to model uncertainties. In Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Detroit, MI, USA, 1–5 October 2023; pp. 8192–8199.
36. Wasiela, S.; Robuffo Giordano, P.; Cortés, J.; Simeon, T. A sensitivity-aware motion planner (samp) to generate intrinsically-robust trajectories. In Proceedings of the 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May–2 June 2023; pp. 12707–12713.
37. Afifi, A.; Belvedere, T.; Pupa, A.; Robuffo Giordano, P.; Franchi, A. Safe and Robust Planning for Uncertain Robots: A Closed-Loop State Sensitivity Approach. *IEEE Robot. Autom. Lett.* **2024**, *9*, 9962–9969.
38. Åström, K.J.; Canudas-de Wit, C. Revisiting the LuGre friction model. *IEEE Control Syst. Mag.* **2008**, *28*, 101–114.
39. Goyal, S.; Ruina, A.; Papadopulous, J. Planar sliding with dry friction, part I. *Wear* **1991**, *143*, 307–330.
40. Canudas de Wit, C.; Olsson, H.; Astrom, K.; Lischinsky, P. A new model for control of systems with friction. *IEEE Trans. Autom. Control* **1995**, *40*, 419–425. https://doi.org/10.1109/9.376053.
41. Johannes, V.; Green, M.; Brockley, C. The role of the rate of application of the tangential force in determining the static friction coefficient. *Wear* **1973**, *24*, 381–385. https://doi.org/10.1016/0043-1648(73)90166-X.
42. Costanzo, M.; De Maria, G.; Natale, C. Two-Fingered In-Hand Object Handling Based on Force/Tactile Feedback. *IEEE Trans. Robot.* **2020**, *36*, 157–173. https://doi.org/10.1109/TRO.2019.2944130.
43. Lynch, K.M.; Park, F.C. *Modern Robotics: Mechanics, Planning, and Control*, 1st ed.; Cambridge University Press: New York, NY, USA, 2017.
44. Waltersson, G.A.; Karayiannidis, Y. Planar Friction Modeling with LuGre Dynamics and Limit Surfaces. *IEEE Trans. Robot.* **2024**, *40*, 3166–3180.
45. Costanzo, M. Control of robotic object pivoting based on tactile sensing. *Mechatronics* **2021**, *76*, 102545. https://doi.org/10.1016/j.mechatronics.2021.102545.
46. Lee, S.H.; Cutkosky, M.R. Fixture planning with friction. *J. Eng. Ind.* **1991**, *113*, 320–327.
47. Howe, R.D.; Cutkosky, M.R. Practical Force-Motion Models for Sliding Manipulation. *Int. J. Robot. Res.* **1996**, *15*, 557–572. https://doi.org/10.1177/027836499601500603.
48. Grüne, L.; Pannek, J. *Nonlinear Model Predictive Control: Theory and Algorithms*; Springer-Verlag: London, United Kingdom, 2011. https://doi.org/10.1007/978-0-85729-501-9.
49. Prattichizzo, D.; Trinkle, J. Grasping. In *Proceedings of the Handbook on Robotics*; Siciliano, B., Kathib, O., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 671–700.
50. Murray, R.M.; Li, Z.; Sastry, S.S. *A Mathematical Introduction to Robotic Manipulation*; CRC Press: Boca Raton, FL, USA, 2017.
51. Costanzo, M.; De Maria, G.; Natale, C.; Pirozzi, S. Design and Calibration of a Force/Tactile Sensor for Dexterous Manipulation. *Sensors* **2019**, *19*, 966. https://doi.org/10.3390/s19040966.
52. Tremblay, J.; To, T.; Sundaralingam, B.; Xiang, Y.; Fox, D.; Birchfield, S. Deep Object Pose Estimation for Semantic Robotic Grasping of Household Objects. In *Proceedings of the 2nd Conference on Robot Learning, Zürich, Switzerland, 29–31 October 2018*;

Billard, A., Dragan, A., Peters, J., Morimoto, J., Eds.; Proceedings of Machine Learning Research; PMLR: New York, NY, USA, 2018; Volume 87, pp. 306–316.

53. Costanzo, M.; De Simone, M.; Federico, S.; Natale, C. Non-Prehensile Manipulation Actions and Visual 6D Pose Estimation for Fruit Grasping Based on Tactile Sensing. *Robotics* **2023**, *12*, 92. https://doi.org/10.3390/robotics12040092.
54. Nicodemus, J.; Kneifl, J.; Fehr, J.; Unger, B. Physics-informed Neural Networks-based Model Predictive Control for Multi-link Manipulators. *IFAC-PapersOnLine* **2022**, *55*, 331–336. https://doi.org/10.1016/j.ifacol.2022.09.117.
55. Habich, T.L.; Mohammad, A.; Ehlers, S.; Bensch, M.; Seel, T.; Schappler, M. Generalizable and Fast Surrogates: Model Predictive Control of Articulated Soft Robots Using Physics-Informed Neural Networks. *IEEE Trans. Robot.* **2025**, *42*, 619–636. https://doi.org/10.1109/TRO.2025.3631818.
56. Norton, R.L. *Kinematics and Dynamics of Machinery*, 1st ed.; McGraw-Hill Education: New York, NY, USA, 2009.
57. Uicker, J.J.; Pennock, G.R.; Shigley, J.E. *Theory of Machines and Mechanisms*, 5th ed.; Oxford University Press: New York, NY, USA, 2016.
58. Dame, A.; Marchand, E. Second-Order Optimization of Mutual Information for Real-Time Image Registration. *IEEE Trans. Image Process.* **2012**, *21*, 4190–4203. https://doi.org/10.1109/TIP.2012.2199124.
59. Zhan, X.; Liu, Y.; Zhu, J.; Li, Y. Homography Decomposition Networks for planar object tracking. *Proc. Conf. AAAI Artif. Intell.* **2022**, *36*, 3234–3242.