Robust Nonprehensile Dynamic Object Transportation: A Closed-loop Sensitivity Approach

Ainoor Teimoorzadeh[1], Andrea Pupa[2], Mario Selvaggio[3] and Sami Haddadin[4]

***Abstract*— In this paper, we propose a closed-loop sensitivity-based approach to enhance the robustness of robotic nonprehensile dynamic manipulation tasks. The proposed method aims at fulfilling the transportation of an object, that is free to move on a tray-shaped robot end-effector, in face of not perfectly known nominal dynamic parameters. The approach is built up on taking the parameterized reference trajectory to be tracked as the optimization variable minimizing a norm of the task closed-loop sensitivity. The resulting optimal reference trajectory is inherently more robust to the parametric variations of object dynamic properties compared to a baseline trajectory execution. The tracking performance is assessed and validated along hardware experiments and an extensive simulation campaign assessing the superior robustness of our approach.**

I. Introduction

Service robots encompass a wide range of useful applications in our daily lives, enhancing efficiency in domestic settings and improving the quality of life especially for the elderly to whom they provide companionship, health monitoring, and assistance with many tasks [1]. Most service robots are, however, still equipped with basic grippers that limit them to clumsy pick-and-place tasks [2]. Dynamic nonprehensile manipulation, on the other hand, allows robots to manipulate objects without requiring any form- or force-closure [3], enabling the use of simpler end-effectors to handle objects that are difficult to be grasped. This form of manipulation expands the realm of application for service robots by avoiding them to impose the bilateral constraints proper of grasping-based manipulation [4]; it additionally improves the manipulation versatility, allowing robots to perform human-like actions such as pushing [5], throwing [6], and rolling [7].

The research leading to these results has been supported by the ARIEL project, in the frame of the PRIN 2022 research program, grant n. 2022WS29WP, funded by the European Union Next-Generation EU, the DARC project, in the frame of the PRIN 2022 PNRR research program, grant n. P2022MHR5C, funded by the European Union Next-Generation EU, and the Lighthouse Initiative Geriatronics by StMWi Bayern (Project X, grant 5140951), LongLeif GaPa gGmbH (ProjectY, grant 5140953). The authors are solely responsible for its content.

[1] A. Teimoorzadeh is with Munich Institute of Robotics & Machine Intelligence, Technical University of Munich (TUM), Germany. E-mail: ainoor.teimoorzadeh@tum.de

[2] A. Pupa is with the Department of Sciences and Method for Engineering, University of Modena and Reggio Emilia, Italy. E-mail: andrea.pupa@unimore.it

[3] M. Selvaggio is with PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, 80125, Naples, Italy. E-mail: mario.selvaggio@unina.it

[4] S. Haddadin is with Mohamed Bin Zayed University of Artificial Intelligence, Masdar City, Abu Dhabi, UAE. E-mail: sami.haddadin@mbzuai.ac.ae

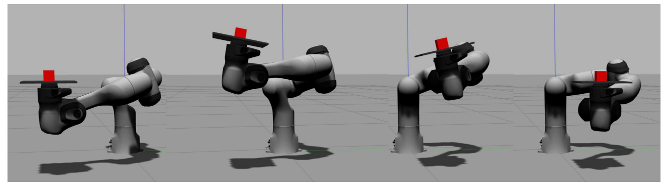

Fig. 1. Snapshots of the main result achieved in this paper: the nonprehensile dynamic transportation of an object from its start (leftmost) to its goal (rightmost) configuration is executed via a tray-like end-effector of a robot following the trajectory that is most robust with respect to object dynamic parameters uncertainty.

Friction-based manipulation enables robots to transport objects along arbitrary trajectories, increasing their capacity for complex tasks like delivering food using a tray-like end-effector [8], [9]. Recently, a shared-control teleoperation architecture with sliding prevention was introduced in [8], where adaptive tray tilting reduces the risk of sliding [10]. Optimal trajectory planning has been explored via quadratic programming [11] and nonlinear model predictive control [12] to enhance tracking performance. Methods that adapt to inaccuracies in friction models while maintaining task speed have been proposed in [13]. However, the parametric uncertainties of the manipulated object dynamics may still significantly affect the control loop performance in this scenario.

The process of designing proper manipulation controllers to enforce a safe and robust service robot behaviour classically relies on the robot and the environment models. Given that the world of mathematical models is distinct from that of the physical systems, a model that perfectly imitates the true physical plant can never be constructed [14]. In fact, 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.

One well-known approach to handle uncertainties is online parameter estimation within the adaptive control framework [15]. However, this requires "persistency of excitation" in robot trajectories, which can degrade task performance. Passivity-based methods offer inherent robustness by handling the disturbance, but dealing with uncertainties often requires complex control laws [16], [17]. Robust control techniques, like H_∞ optimization, must be designed carefully to achieve certain performance criteria in presence of uncertainties [14].

Alternatively, one can plan and execute trajectories that minimize the effect of uncertainties, inherently robustifying the robot behavior [18]–[22]. These methods usually do not

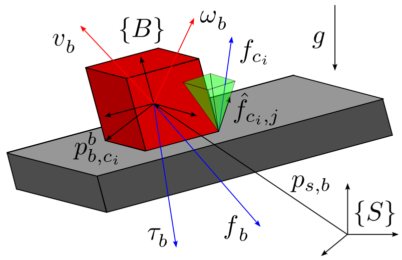

Fig. 2. Drawing of the tray-based non-prehensile manipulation model: the object (in red) is manipulated via a tray-based robotic end–effector (in grey). Symbols are described in Sec. II.

take into account the controller, or require specific controller designs. In this context, 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 [23]. 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 [24], energy tank initialization for manipulators [25], and path planning [26]. 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 [27]. However, none of the previous works has ever considered using closed-loop sensitivity to robustly handle parametric uncertainties of the manipulated object in a nonprehensile manipulation scenario.

With the aim of filling this gap in the nonsliding dynamic manipulation literature, the main goal of this paper is to provide a control-aware dynamic planning framework able to solve the problem of computing the most robust rest-to-rest trajectory that brings the object from point A to point B. In other words, this practically means computing trajectories whose execution is less sensitive to parametric variations of the object dynamic properties.

The paper is structured as follows. The system modelling and the preliminaries are discussed in Sec. II. The closed-loop sensitivity approach is explained in Sec. III in a nutshell. The proposed controller is discussed in Sec. IV. The resulting control is applied to a robot manipulator, and the simulation results of the closed-loop are shown in Sec. V. Conclusions and future works are presented in Sec. VI.

II. System Modeling

The model of the tray-based nonprehensile transportation system, depicted in Fig. 2, is briefly described in the following. We denote with $q_{s,b} = (p_{s,b}, R_{s,b}) \in \mathrm{SE}(3)$ the current pose of the object frame $\{B\}$, fixed to the object center of mass, in the inertial reference frame $\{S\}$, where $p_{s,b} \in \mathbb{R}^3$ denotes the object position vector, and $R_{s,b} \in \mathrm{SO}(3)$ its orientation. Hereafter, unless otherwise stated, all the quantities are express in their body frame. The object dynamics can be written in body coordinates as [28]

$$M_b \dot{\mathcal{V}}_b + C_b(\mathcal{V}_b)\mathcal{V}_b + N_b\,(R_{s,b}) = \mathcal{F}_b, \tag{1}$$

where $M_b \in \mathbb{R}^{6\times6}$ is the constant and positive-definite object mass matrix constructed from the object mass $m > 0$ and the constant symmetric inertia matrix $I \in R^{3\times3}$; $C_b \in \mathbb{R}^{6\times6}$ is the matrix accounting Centrifugal/Coriolis effects; $N_b\,(R_{s,b}) \in \mathbb{R}^6$ encodes the gravity force, $\mathcal{V}_b = (v_b,\, \omega_b) \in \mathbb{R}^6$ is the object twist, with $v_b,\, \omega_b \in \mathbb{R}^3$ linear and angular velocity vectors, respectively; $\mathcal{F}_b = (f_b,\, \tau_b) \in \mathbb{R}^6$ is the applied wrench, with $f_b,\, \tau_b \in \mathbb{R}^3$ force and torque vectors, respectively, all specified with respect to $\{B\}$.

We assume the body wrench $\mathcal{F}_b$ being realized by means of a set of wrenches exerted at the $n_c = 4$ contact points located along the object perimeter. The pose of the contact points in $\{B\}$ are known and correspond with the 4 object vertexes in contact with the tray. The $i-$th contact point is identified by a contact frame $\{C_i\}$ whose pose in $\{B\}$ is expressed by $q_{b,c_i} = (p_{b,c_i}, R_{b,c_i}) \in \mathrm{SE}(3)$.

To mathematically describe the object/tray interaction behavior, a suitable contact model is now introduced. We assume all the contacts behave according to the point contacts with friction model [29], i.e. only linear forces $f_{c_i} \in \mathbb{R}^3$, can be transmitted through the $i-$th contact. This allows expressing $\mathcal{F}_b$ as

$$\mathcal{F}_b = G F_c, \quad \text{with} \quad G = \left[\mathrm{Ad}^{\mathrm{T}}_{q^{-1}_{b,c_1}} B_c, \ldots, \mathrm{Ad}^{\mathrm{T}}_{q^{-1}_{b,c_{n_c}}} B_c\right], \tag{2}$$

where $G \in \mathbb{R}^{6\times 3n_c}$ maps the stacked vector of independent contact forces $F_c = [f^{\mathrm{T}}_{c_1}, \ldots, f^{\mathrm{T}}_{c_{n_c}}]^{\mathrm{T}} \in \mathbb{R}^{3n_c}$ to the body wrench $\mathcal{F}_b$ exerted at the object center of mass, the symbol $\mathrm{Ad}^{\mathrm{T}}_{q^{-1}_{b,c_i}}$ expresses the adjoint matrix corresponding to the transformation q^{-1}_{b,c_i}, while the matrix $B_c \in \mathbb{R}^{6\times3}$ 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 write as follows

$$\mathrm{Ad}^{\mathrm{T}}_{q^{-1}_{b,c_i}} = \begin{bmatrix} R_{b,c_i} & 0 \\ -R_{b,c_i}\hat{p}_{c_i,b} & R_{b,c_i} \end{bmatrix}, \quad \text{and} \quad B_c = \begin{bmatrix} I_{3\times3} \\ 0_{3\times3} \end{bmatrix} \tag{3}$$

where $\hat{p}_{b,c_i} \in \mathrm{so}(3)$ denotes the skew-symmetric matrix associated with the vector $p_{b,c_i} \in \mathbb{R}^3$, i.e. the position of the $i-$th contact point expressed in $\{B\}$.

Another essential component of the contact model is the *friction cone* defined at the $i-$th contact as the set of generalized contact forces that can be physically realized. In this work, we use a linearly approximated representation of the composite friction cone space $FC = FC_{c_1} \times \cdots \times FC_{c_{n_c}} \subset \mathbb{R}^{3n_c}$, where each FC_{c_i} is conservatively approximated with a cone-inscribed polyhedral generated by k vectors $\hat{f}_{c_i}$. As shown in Fig. 2, when $k = 4$ the circular cone is approximated by a pyramid. This is a common and conservative approximation which allows to treat the (otherwise second order) friction cone as a linear constraint in the following optimization problem. Indeed, the condition $f_{c_i} \in FC_i$ can be conveniently formulated expressing f_{c_i} as a nonnegative linear combination of unit vectors $\hat{f}_{c_i,1}, \ldots, \hat{f}_{c_i,k}$ i.e.

$$FC_{c_i} = \left\{ f_{c_i} \in \mathbb{R}^3 \,\middle|\, f_{c_i} = \sum_{j=1}^{k} \lambda_{c_i,j} \hat{f}_{c_i,j},\ \lambda_{c_i,j} \geq 0 \right\}. \quad (4)$$

The set of feasible contact forces are those belonging to the linearized friction cone space in (4) and can be compactly written as

$$F_c = \hat{F}_c \Lambda, \quad \text{with} \quad \Lambda = [\lambda_{1,1}, \ldots, \lambda_{n_c,k}]^{\mathrm{T}} \geq 0, \quad (5)$$

where the inequality is intended component-wise and

$$\hat{F}_c = \operatorname{diag}\left(\hat{F}_{c,1}, \ldots, \hat{F}_{c,n_c}\right), \quad \hat{F}_{c,i} = \left[\hat{f}_{c,1}, \ldots, \hat{f}_{c,k}\right].$$

In summary, substituting F_c from (5) in (2) yields

$$\mathcal{F}_b = G\hat{F}_c\Lambda, \quad \text{with} \quad \Lambda \geq 0.$$

According to the mechanics of physical contacts, contact forces feasibility impose that each component of the vector Λ can be either positive or zero. When all the components are positive, friction forces are entirely inside the friction cone space, making the object stationary to the tray. As some components approach zero, the corresponding contacts might be sliding. A nonsliding manipulation controller for the system in (1) must thus choose Λ to steer the state to a desired one, without violating its non-negativity constraint. However, a model-based non-sliding controller designed using the nominal object dynamic parameters can still generate sliding when the real parameters differ from the nominal ones. The goal of this paper is to find object trajectories whose tracking exhibit the minimum variation of object state $x = (q_{s,b}, \mathcal{V}^b)$ and input $u = \Lambda$ in face of uncertainties. In this way, we prevent large performance degradation of the non-sliding controller, thus also minimizing the possibility to slide, enhancing the robustness of the execution.

III. Closed-loop Sensitivity

To realize our goal, we exploit the notion of *closed-loop sensitivity* summarized in the following. Consider the generic system dynamics

$$\dot{x} = f(x, u), \quad (6)$$

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 possibly uncertain parameters. For instance, in our case (1), one may account for uncertainties in the manipulated object's mass, inertia, or its friction parameter. 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

$$\begin{cases} \dot{\xi} = g(\xi, x, y_d(t), p_n, t), \\ u = h(\xi, x, y_d(t), p_n, t). \end{cases} \quad (7)$$

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 parameters).

For the closed-loop system it is possible to define the *state sensitivity matrix* as

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

and the *input sensitivity matrix* as

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

that 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

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

Matrices Π and Θ can be used for several purposes, e.g., to generate the reference trajectory $y_d(t)$ that produces less sensitive behaviours [23], or to evaluate the tubes (or ellipsoids) of enveloping uncertain states/inputs [27]. Assuming that each parameter p_i can vary in a given range of δp_i, centered at a nominal p_{c_i}, i.e.

$$p_i \in [p_{c_i} - \delta p_i,\ p_{c_i} + \delta p_i], \quad (8)$$

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

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

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

$$\Delta x^T (\Pi W \Pi^T)^{-1} \Delta x = 1, \quad (9)$$

where x_c is the state evolution of (6) and (7) in the unperturbed case ($p = p_c$), and in the input space

$$\Delta u^T (\Theta W \Theta^T)^{-1} \Delta u = 1, \quad (10)$$

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 (10) for obtaining the tubes of perturbed trajectories for the individual components of the inputs. For each direction of interest n_i in the input space, one can obtain the "tube radius" r_i as

$$r_i = \sqrt{n_i^T \Theta W \Theta^T n_i}. \quad (11)$$

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

$$u_{c,i} - r_i \leq u_i \leq u_{c,i} + r_i, \quad (12)$$

with $u_{c,i}$ being the behaviour of the i-th input in the unperturbed case ($p = p_c$). Equation (12) bounds all the perturbed inputs when the parameter uncertainty is bounded as in (8). The same approach can be applied to compute the envelope of the perturbed states x, or of any quantity whose sensitivity matrix is available.

A. Optimization Problem

Thanks to the construction of tubes, state-of-the-art approaches [24], [27] propose to use sensitivity to avoid actuators 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 showed in [12], it is possible to build a trajectory $y^*(t) = \left(q^*_{s,b_0}, \mathcal{V}^*_{b_0}, \dot{\mathcal{V}}^*_{b_0}\right)$ for the system guaranteeing that the object will not slide, i.e. $\Lambda \geq 0$. However, in the presence of parameter uncertainty, e.g. the weight of the mass, tracking this optimal trajectory may still generate negative value of Λ, causing the sliding of the mass.

To avoid this, it is possible employ a proper trajectory optimization problem. Firstly, the desired trajectory can be expressed as a 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 along it. Formally

$$a = \arg\min \int_0^T ||\Pi(t)||^2 \mathrm{d}\tau \tag{13}$$

$$\text{s.t.} \quad y^*(a, t_0) = y_0 \tag{14}$$

$$y^*(a, t_f) = y_f \tag{15}$$

$$\Lambda(t) - r_\Lambda(t) > 0 \qquad \forall t \in [t_0, t_f] \tag{16}$$

$$\dot{x} = f(x, u) \tag{17}$$

where $y_0 = \left(q_{s,b_0}, \mathcal{V}_{b_0}, \dot{\mathcal{V}}_{b_0}\right)$ and $y_f = \left(q_{s,b_f}, \mathcal{V}_{b_f}, \dot{\mathcal{V}}_{b_f}\right)$ represent the initial and final conditions, respectively, and $r_{\Lambda_i}(t)$, represents the tube radius of each component λ_i at the time instant t, computed via (11).

The first and second constraints guarantee that the optimal trajectory 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 nonprehensile transportation. Indeed, thanks to the definition of the tubes, it is possible to guarantee that even in the worst case scenario, i.e. the lowest perturbed input, the contact forces stay in the friction cone. The last constraint, instead, represents the dynamic evolution of the system.

In summary, the goal of the optimization problem in (13) 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 behaviour will be as robust as possible; $ii)$ the control action is always feasible, even in the presence of parameter uncertainties.

IV. Control

In this section, we describe the controller for our object-tray system in Sec. IV-A, and explain how this is practically implemented via a robot manipulator controller in Sec. IV-B.

A. Object Controller

To proceed with the description of the object controller, we first express the system dynamics in (1) in the form of (6), as also requested in Sec III. Since we chose $x = (q_{s,b}, \mathcal{V}_b)$ and $u = \Lambda$, the system dynamics can be written as follows

$$\dot{x} = \begin{cases} \dot{q}_{s,b} = \dot{q}_{s,b} \\ \dot{\mathcal{V}}_b = M_b^{-1}(q)\left(G\hat{F}_c u - C_b(\mathcal{V}_b)\mathcal{V}_b - N_b\left(R_{s,b}\right)\right) \end{cases}. \tag{18}$$

Taking inspiration by previous works on the topic (e.g. [12]), a suitable control law to track a desired trajectory $y^*(t) = \left(q^*_{s,b}, \mathcal{V}^*_b, \dot{\mathcal{V}}^*_b\right)$ can be devised as follows

$$\begin{aligned}\Lambda = \left(G\hat{F}_c\right)^\dagger \Big(C_b(\mathcal{V}_b)\mathcal{V}_b + N_b\left(R_{s,b}\right) + M_b\Big(\dot{\mathcal{V}}^*_b + \\ + K_d\left(\mathcal{V}^*_b(t) - \mathcal{V}_b\right) + K_p\left(q^*_{s,b} - q_{s,b}\right)\Big)\Big).\end{aligned} \tag{19}$$

Substituting (19) in (18), it is trivial to show that this control achieves the asymptotic tracking of the desired time varying trajectory imposing the following system dynamics

$$\dot{\mathcal{V}}^*_b - \dot{\mathcal{V}}_b + K_d\left(\mathcal{V}^*_b - \mathcal{V}_b\right) + K_p\left(q^*_{s,b} - q_{s,b}\right) = 0.$$

Note that the use of the Moore-penrose pseudoinverse of $G\hat{F}_c$ in (19) results from the minimization of the two-norm of Λ when the constraint $\Lambda \geq 0$ is not binding.

B. Robot Manipulator Controller

The optimal trajectory, solution of (13), would provide a robust task execution exploiting the object controller in Sec. IV-A, as long as a torque-controlled manipulator would be able to realize the object motion. The objective now is to find the manipulator generalized control forces that realize our goal. Assuming that the tray dynamics is compensated via suitable augmentation of the last link dynamic parameters, the dynamics of the n-DoF ($n \geq 6$) manipulator can be written in joint coordinates $\theta \in \mathbb{R}^n$ as

$$M_r(\theta)\ddot{\theta} + C_r(\theta, \dot{\theta})\dot{\theta} + N_r(\theta) = \tau - J_c^{\mathrm{T}}(\theta)F_c \tag{20}$$

where $M_r(\theta) \in \mathbb{R}^{n\times n}$ is the positive-definite joint-space inertia matrix, $C_r(\theta, \dot{\theta}) \in \mathbb{R}^{n\times n}$ is the Coriolis/Centrifugal matrix, while $N_r(\theta) \in \mathbb{R}^n$ stands for the gravity term, $\tau \in \mathbb{R}^n$ is the vector of the generalised joint control input, and $J_c \in \mathbb{R}^{3n_c\times n}$ encodes the body Jacobian of the contact points. The $i-$th block row of J_c can be computed resorting to the definition of the adjoint matrix given in (3) as

$$J_{c_i}(\theta) = \mathrm{Ad}_{g_{c_i,e}} J_e(\theta), \tag{21}$$

in which $J_e(\theta) \in \mathbb{R}^{6\times n}$ is the manipulator body Jacobian of the tray end–effector. The control torque τ can be designed resorting to a computed torque control paradigm as [2]

$$\tau = J_c^{\mathrm{T}}(\theta)F_c + C_r(\theta, \dot{\theta})\dot{\theta} + N_r(\theta) + M_r(\theta)u. \tag{22}$$

Substituting (22) in (20), leads to $u = \ddot{\theta}$. At this point, to design u, we start from the linear relation between the object space velocities and the joint space ones, given by the object body Jacobian $J_b \in \mathbb{R}^{6\times n}$ through $\mathcal{V}_b = J_b\dot{\theta}$. Differentiating this leads to

$$\dot{\mathcal{V}}_b = J_b\ddot{\theta} + \dot{J}_b\dot{\theta}. \tag{23}$$

We then exploit the *fundamental grasp constraint* that holds when $F_c \in FC$ (see [28] for more details)

$$J_c(\theta)\dot{\theta} = G^{\mathrm{T}}\mathcal{V}_b. \tag{24}$$

Solving (24) for $\dot{\theta}$, and substituting this in (23), suggests the choice of the control law

$$u = J_b^{\dagger}(\dot{\mathcal{V}}_b^{*} - \dot{J}_b\dot{\theta}) + N\dot{\theta}_0, \tag{25}$$

where $N = I - J_b^{\dagger}J^b \in \mathbb{R}^{n\times n}$ is the null-space projector and $\dot{\theta}_0$ is a secondary task velocity control input [2]. The controller guarantees the execution of the desired trajectory via the realization of the desired object accelerations $\dot{\mathcal{V}}_b^{*}$ although it is subjected to the inertial and gravity forces.

V. Simulation and Hardware Validation

We now proceed detailing the simulation and hardware experiments carried out to validate the closed-loop sensitivity nonprehensile manipulation framework presented so far. We consider a cuboid object of the dimensions of $4 \times 4 \times 4$ cm to be transported from the initial position of $p_{s,b} = (0.54237, -0.370574, 0.4)^T$ m to the final position of $p_{s,b} = (0.444272, 0.100574, 0.4)^T$ m. The object is held by a $3D$ printed 20×20 cm tray which is mounted in the upright configuration on a 7-DOF Franka Emika Panda robot arm end-effector and is left free to slide with Coulomb friction coefficient equal to 0.5. The inertial parameters of the tray, as well as the connecting screws, have been included in the robot model to ensure the parametric variation is restricted to the object variations. The parameters of the simulation/experimental setup are provided in Table I.

The constrained minimization problem in (13) is formulated and implemented using CasADi in MATLAB [30] and has been numerically evaluated imposing the initial and final trajectory values (configurations and their time derivatives representing twist and acceleration) for $y^*(a,t)$[1]. Here a denotes the vector of trajectory coefficients. We have encoded these constraints in the form

$$Ma = d. \tag{26}$$

The homogeneous solution of (26), defined by $a = M^{\dagger}d$, has been taken as the initial guess of the optimization problem as

[1]The positional part of the trajectory is parametrized using polynomials. For rotations, polynomials are applied to the angular velocities, and the final orientation is obtained through their forward integration.

TABLE I

Dynamic and control parameters

Tray	$m = 0.150$ [Kg]	$I = \mathrm{diag}(1.25^{-4})$ [Kgm^2]
Manipulated object nominal values	$m_n = 0.256$ [Kg]	$I_n = \mathrm{diag}(3.4^{-4})$ [Kgm^2]
Manipulated object max perturbation	$\pm 10\% m_n$ [Kg]	$I_r = \mathrm{diag}(3.4^{-4})$ [Kgm^2]
Control Parameters	K_p = diag(500, 300)	K_d = diag(25, 15)

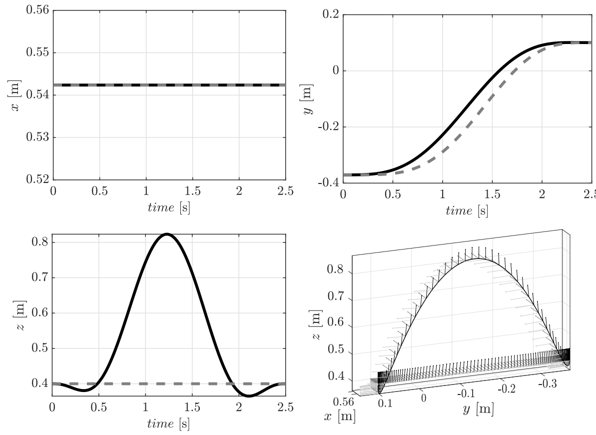

Fig. 3. Time evolution of the robust trajectory (solid black), and of the baseline trajectory (dashed gray) and their 3D visualization.

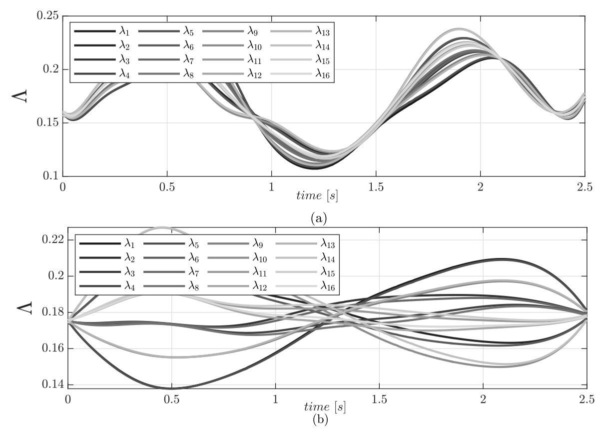

Fig. 4. Time history of Λ with nominal object mass/inertial parameters: (a) robust trajectory, (b) baseline trajectory.

also done in [23]. The optimization problem in (13) required 16.74 s to be solved, which is suitable for offline trajectory planning. We use a linear trajectory connecting the start and the end configurations as our initial value. The resulting robust trajectory, shown in Fig. 3, illustrates that the optimal path, aimed at achieving a displacement of 0.4 m along the y-axis, predominantly induces a bell-shaped pattern in the z-axis, independent of the selected t_f, when compared to a our linear baseline trajectory generated for the same duration. Intuitively, this is because, by accelerating upwards, the normal component of the contact forces increases, with a consequent increment of the binding effect. Figure 4, depicts the tray's reaction force coefficients Λ for both cases, indicating that they adhere to the necessary non-sliding conditions imposed by the linearized friction model. Although lower values are achieved in the optimal case, this solution is supposed to exhibit less variation in face of parameters uncertainty.

To demonstrate this, we validate the effectiveness of the computed sensitivity tubes in capturing the input evolution under parametric uncertainties through extensive simulations, comparing the robust trajectory with the baseline one. To this end, we have performed 100 simulations in the Gazebo physics engine simulating the Franka Emika Panda arm via

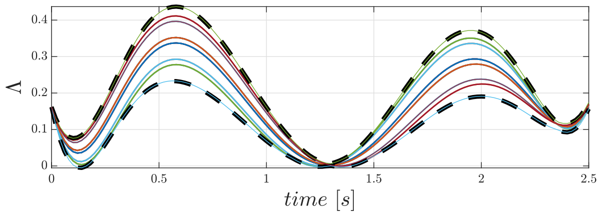

Fig. 5. Time history of some Λ upon perturbations in the nominal mass of the object considering the robust trajectory. The black dashed lines illustrate the envelope of all the perturbed trajectories input, i.e. the minimum and maximum of each tube.

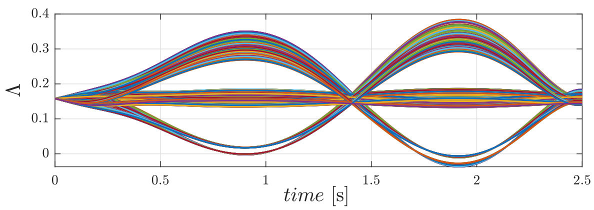

Fig. 6. Time history of Λ upon perturbations in the nominal mass of the object considering the baseline trajectory.

the franka-ROS interface, varying the inertial parameters of the cuboid up to 10% of their nominal values. The initial and final pose, the boundary conditions, as well as the duration of the trajectory have been kept constant and equal to 2.5 s throughout all the simulations. Figure 5 represents the time evolution of the control inputs for optimal case in view of the applied perturbations with the tubes. Due to the high numbers of control inputs, for clarity only the minimum and the maximum of all tubes are presented, together with the evolution of some of the perturbed inputs. The results of the simulations show that the tubes can successfully capture the envelope of the control inputs while respecting the necessary condition for the non-sliding manipulation. The only exception is around $t = 0.15$ s, where the tubes slightly go below 0. This is due to the constraint tolerance of the nonlinear optimization solver. On the contrary, Figure 6, highlights the fact that a linear trajectory is only capable of maintaining the desired behavior under nominal conditions, while it fails to uphold these in the perturbed scenarios. Indeed, the value of Λ becomes significantly negative in some portion of the trajectory (roughly between 1.7 and 2.1 s), denoting the sliding of the object.

To further validate the aforementioned conclusion, we replicated simulation results performing experiments on the hardware setup (see Fig. 7). To track the object pose, the Vicon motion capture system has been used. Four markers were located on the object face opposed to the tray and four markers in the corners of the tray. The tracking performance was analyzed during the execution of both trajectories, as illustrated by the snapshots in Fig. 7. Figure 8, illustrates the time history of the tracking error norm for the robust trajectory (in black) and the baseline trajectory (in gray) under both the nominal scenario (a) and a case of mass perturbation equal to 50 gr (b). It can be noted that, the same deviation of the nominal mass generates a bigger variation of the error in the baseline case (in gray). This demonstrates that

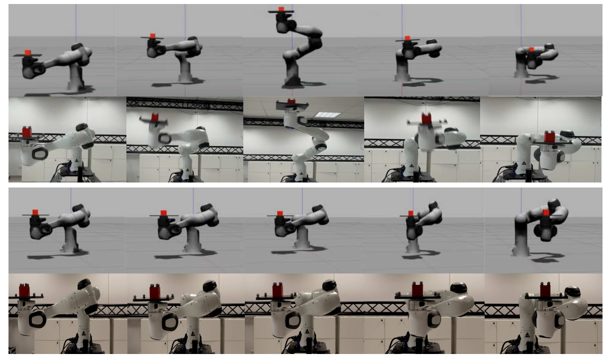

Fig. 7. Snapshots of the robust (top) and the baseline (bottom) trajectory executed in simulation and with the experimental hardware setup.

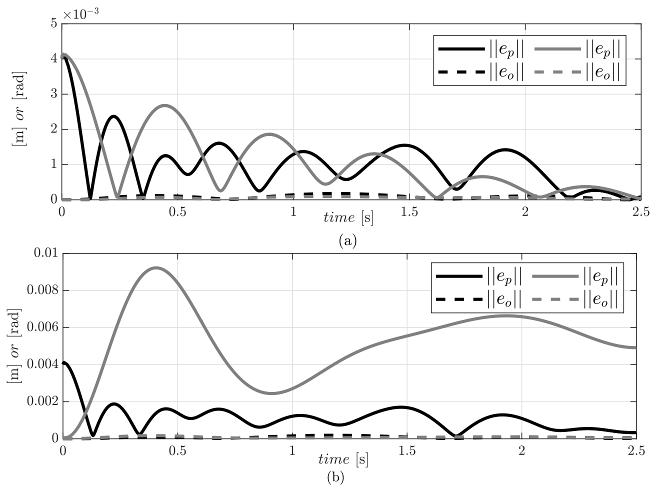

Fig. 8. Time history of the norm of the tracking pose error for the nominal scenario (a) and the perturbed case (b), along the robust trajectory (in black) and the baseline trajectory (in gray). e_p, e_o are the position and orientation errors, respectively.

the baseline trajectory is more sensitive to such variations, leading more easily to constraint violation and sliding when perturbations are present. The results can also be appreciated in the accompanying video, which provides more insights and results not inserted here due to limitation of space.

VI. CONCLUSIONS

In this paper, we have proposed to use a close-loop sensitivity approach to robustify the execution of dynamic non-prehensile object transportation tasks performed by robotic manipulators. The followed approach showed that the minimization of the sensitivity norm provides a simple and efficient computational framework to compute trajectories whose execution is minimally perturbed by uncertainties in the model dynamics, without the need of complex control artifacts. We validated our framework along simulation and hardware experiments showing the superior performance of the task execution along the optimized trajectory when compared to baseline straight segment paths. As for the future developments, we plan to extend the framework including more general object and contact models and their corresponding uncertainties (e.g. friction coefficient).

References

[1] M. Tröbinger, C. Jähne, Z. Qu, J. Elsner, A. Reindl, S. Getz, T. Goll, B. Loinger, T. Loibl, C. Kugler *et al.*, "Introducing garmi-a service robotics platform to support the elderly at home: Design philosophy, system overview and first results," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5857–5864, 2021.

[2] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.

[3] A. Bicchi, "On the closure properties of robotic grasping," *The International Journal of Robotics Research*, vol. 14, no. 4, pp. 319–334, 1995.

[4] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1711–1718, 2018.

[5] F. Bertoncelli, F. Ruggiero, and L. Sabattini, "Linear time-varying mpc for nonprehensile object manipulation with a nonholonomic mobile robot," in *IEEE International Conference on Robotics and Automation*, 2020, pp. 11 032–11 038.

[6] A. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, "Coordinate-free framework for robotic pizza tossing and catching," in *IEEE International Conference on Robotics and Automation*, 2016, pp. 3932–3939.

[7] D. Serra, F. Ruggiero, A. Donaire, L. Buonocore, V. Lippiello, and B. Siciliano, "Control of nonprehensile planar rolling manipulation: A passivity-based approach," *IEEE Trans. Robot.*, vol. 35, no. 2, pp. 317–329, 2019.

[8] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero, and P. Robuffo Giordano, "A shared-control teleoperation architecture for nonprehensile object transportation," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 569–583, 2021.

[9] J. Miguel Garcia-Haro, S. Martinez, and C. Balaguer, "Balance computation of objects transported on a tray by a humanoid robot based on 3d dynamic slopes," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–6.

[10] G. Martucci, J. Bimbo, D. Prattichizzo, and M. Malvezzi, "Maintaining stable grasps during highly dynamic robot trajectories," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9198–9204.

[11] R. Subburaman, M. Selvaggio, and F. Ruggiero, "A non-prehensile object transportation framework with adaptive tilting based on quadratic programming," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3581–3588, 2023.

[12] M. Selvaggio, A. Garg, F. Ruggiero, G. Oriolo, and B. Siciliano, "Non-prehensile object transportation via model predictive non-sliding manipulation control," *IEEE Transactions on Control Systems Technology*, vol. 31, no. 5, pp. 2231–2244, 2023.

[13] W. Yang and M. Posa, "Dynamic on-palm manipulation via controlled sliding," *arXiv preprint arXiv:2405.08731*, 2024.

[14] K. Zhou and J. C. Doyle, *Essentials of Robust Control*. Prentice Hall, 1998, vol. 104.

[15] K. J. Astrom and B. Wittenmark, "Adaptive control 2nd edition," *Addison-Wesley Pub Co.*, vol. 1994, 1994.

[16] R. Ortega, A. Van Der Schaft, B. Maschke, and G. Escobar, "Interconnection and damping assignment passivity-based control of port-controlled hamiltonian systems," *Automatica*, vol. 38, no. 4, pp. 585–596, 2002.

[17] A. Teimoorzadeh, A. Donaire, P. Arpenti, and F. Ruggiero, "Robust energy shaping for mechanical systems with dissipative forces and disturbances," in *2022 European Control Conference (ECC)*, 2022, pp. 1409–1414.

[18] S. Candido and S. Hutchinson, "Minimum uncertainty robot path planning using a pomdp approach," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 1408–1413.

[19] ——, "Minimum uncertainty robot navigation using information-guided pomdp planning," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 6102–6108.

[20] A. Ansari and T. Murphey, "Minimum sensitivity control for planning with parametric and hybrid uncertainty," *The International Journal of Robotics Research*, vol. 35, no. 7, pp. 823–839, 2016.

[21] R. Tedrake, I. R. Manchester, M. Tobenkin, and J. W. Roberts, "Lqr-trees: Feedback motion planning via sums-of-squares verification," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1038–1052, 2010.

[22] A. Majumdar and R. Tedrake, "Funnel libraries for real-time robust feedback motion planning," *The International Journal of Robotics Research*, vol. 36, no. 8, pp. 947–982, 2017.

[23] P. Robuffo Giordano, Q. Delamare, and A. Franchi, "Trajectory generation for minimum closed-loop state sensitivity," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 286–293.

[24] A. Srour, A. Franchi, and P. Robuffo Giordano, "Controller and trajectory optimization for a quadrotor uav with parametric uncertainty," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 1–7.

[25] A. Pupa, P. Robuffo Giordano, and C. Secchi, "Optimal energy tank initialization for minimum sensitivity to model uncertainties," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 8192–8199.

[26] S. Wasiela, P. Robuffo Giordano, J. Cortés, and T. Simeon, "A sensitivity-aware motion planner (samp) to generate intrinsically-robust trajectories," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 12 707–12 713.

[27] A. Afifi, T. Belvedere, A. Pupa, P. Robuffo Giordano, and A. Franchi, "Safe and robust planning for uncertain robots: A closed-loop state sensitivity approach," *IEEE Robotics and Automation Letters*, vol. 9, no. 11, pp. 9962–9969, 2024.

[28] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.

[29] D. Prattichizzo and J. Trinkle, "Grasping," in *Handbook on Robotics*, B. Siciliano and O. Kathib, Eds. Springer, 2008, pp. 671–700.

[30] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, 2018.