A Non-prehensile Object Transportation Framework with Adaptive Tilting based on Quadratic Programming

Rajesh Subburaman[1], Mario Selvaggio[1,2], and Fabio Ruggiero[1,3]

***Abstract*—This work proposes an operational space control framework for non-prehensile object transportation using a robot arm. The control actions for the manipulator are computed by solving a quadratic programming problem considering the object's and manipulator's kinematic and dynamic constraints. Given the desired transportation trajectory, the proposed controller generates control commands for the robot to achieve the desired motion whilst preventing object slippage. In particular, the controller minimizes the occurrence of object slippage by adaptively regulating the tray orientation. The proposed approach has been extensively evaluated numerically with a 7-degree-of-freedom manipulator, and it is also verified and validated with a real experimental setup.**

***Index Terms*—Dexterous Manipulation; Optimization and Optimal Control; Intelligent Transportation Systems**

I. Introduction

TRANSPORTING objects of different weights and dimensions using a tray constitute a more agile and efficient solution than prehensile grasping-based manipulation. Tray-based object transportation is a non-prehensile manipulation task in which preventing objects' sliding is paramount to successfully carrying them from one place to another. Further, it has huge application potential in various fields, such as a robotic waiter carrying food items in a restaurant, mobile service robots delivering medicines/equipment, mobile robots handling boxes in an industry/warehouse, etc. However, transferring this skill to a robot is not trivial, as they must possess accurate models of how their actions will affect the world. Without this, robots are slow, clumsy, and less robust in performing non-prehensile tray-based transportation tasks. To enable human-like and highly dynamic movements, a robot must consider inertial forces that arise on an object and enforce a non-sliding behavior exploiting frictional constraints [1].

Chronologically, non-prehensile manipulation is a recently emerged alternative to the grasping-based one [2]. Its strength resides in avoiding planning how to approach and grasp an object and subsequently controlling the internal stresses that arise during manipulation. Restricting robots to only grasp objects artificially limits the tasks they can accomplish. Leveraging a larger set of manipulative actions, it is possible to handle objects that are too large, too heavy, or too cluttered to be grasped by a robotic gripper. In this scenario, an object can be manipulated through simpler end-effectors (i.e., a simple palm instead of a dexterous hand) through a sequence of non-prehensile manipulation primitives [3] such as pushing [4], throwing [5], tossing [6], catching [7], rolling [8], etc.

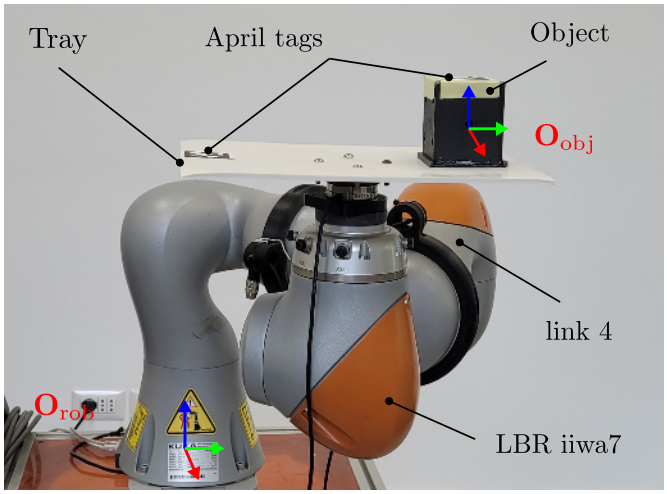

Fig. 1: The non-prehensile object transportation system involves a 7-DoF LBR iiwa7 robot, a tray-like end effector, and a mild steel hollow cuboid as the object to be transported. $\mathbf{O}_{\text{rob}}$, and $\mathbf{O}_{\text{obj}} \in \mathbb{SE}(3)$ represent the coordinate frame attached to the robot's base and object's center of mass respectively. The April tags are used to measure the tray-object relative displacement.

In this work, the object transportation problem has been addressed by considering non-prehensile manipulation principles citing the problem's application potential. In this regard, motion planning methods exist to plan fast and dynamic trajectories for the non-prehensile transportation of a bottle [9] and multiple objects [10]. However, unmodelled effects can cause the plan to fail when implemented on a real robot [11]. Only a few works have proposed control solutions to this problem in the past. In [3], non-prehensile dynamic grasp constraints are enforced using a linear quadratic regulator with saturated outputs and contact forces projected to the feasible contact space. Previous works have proposed to more robustly avoid object slippage by changing the tray orientation. In [12], an offline trajectory optimization method is proposed to prevent slippage while minimizing the task duration. Starting from the linear trajectory, the optimization procedure iteratively finds rotational motions that make contact forces fall inside the spatial grasp force space. The need for online re-planning was highlighted in that work. In [13], the tray orientation is reactively adapted to increase the distance of the contact forces from the friction cone borders, thus being more robust with respect to friction coefficient overestimates. However, if not

Manuscript received: January 13, 2023; Revised March 20, 2023; Accepted April 7, 2023.

This paper was recommended for publication by Editor Hong Liu upon evaluation of the Associate Editor and Reviewers' comments.

The research leading to these results has been supported by the European Union's Horizon 2020 research and innovation program under grant agreement No 101017008 (Harmony).

[1]Authors are with the C.R.E.A.T.E. Consortium, Via Claudio 21, Naples, 80125, Italy. [2]Author is with the ICAROS Centre, University of Naples Federico II, Via Pansini 5, Naples, 80131, Italy. [3]Author is with the PRISMA Lab, Department of Electrical Engineering and Information Technology, University of Naples Federico II, Via Claudio 21, Naples, 80125, Italy. Corresponding author's email: mithram14.rajesh06@gmail.com

Digital Object Identifier (DOI): see top of this page.

adequately accounted for, the robotic system bounds may be violated while realizing the required rotational motion.

Recent works endow humanoids [14] or quadrupeds [15], [16] with tray-based non-prehensile manipulation skills to integrate them into human-centric environments.

In reference to the aforementioned works, this paper proposes a model-based controller for the non-prehensile transportation of an object on a tray of a torque-controlled manipulator (Fig. 1). The controller is devised according to the Operational Space Control (OSC) framework, and its control actions are computed from a Quadratic Problem (QP) solution. Given a desired trajectory for the transported object, the proposed controller generates control commands to achieve the desired motion while enforcing non-sliding conditions for the object and taking into account the kinematic and dynamic constraints of the manipulator. In addition, the controller minimizes the object slippage by adaptively regulating the tray orientation to robustify the performance.

In particular, we bring the following contributions. 1) The previous works [3], [9]–[12] report an offline planned and optimized trajectory. Here, we both generate and optimize the trajectory online with respect to various constraints, making it more reactive and dynamic. 2) Unlike the other works where only the application constraints are considered, this work includes several system constraints which ensure the practical realization of the task on real systems. 3) Dynamic equilibrium-based adaptive tilting function is proposed to realize high object accelerations during non-prehensile transportation minimizing the occurrence of object slippage along the tray. Though [13] reports a similar reactive tilting technique, this was not optimized considering the system limits. 4) An OSC framework is used to realize the motion by exploiting its merits, such as the nonrequirement of inverse kinematics, minimal optimization variables, and dynamically consistent motion.

II. System Description and Modeling

Section II-A describes the considered non-prehensile object transportation system. The models used in this work are dealt with in the rest of the section. Namely, the integrated manipulator-object dynamics in task space is derived in Section II-B, while the contact model and its associated friction constraints are succinctly introduced in Section II-C.

A. System Description

The non-prehensile object transportation system considered here involves a manipulator with n degrees of freedom (DOFs), a tray-like end effector, and a cuboid object to be transported. The complete system is shown in Fig. 1. The following assumptions are considered for modelling the aforementioned system. $(A1)$ The object's kinematic and dynamic properties are assumed to be known and it is considered to have a larger footprint, i.e., the object's width $\geq$ height; $(A2)$ the object's initial pose with respect to the tray is observable; $(A3)$ the tray-object interaction is considered to occur via a fixed number of contact points located on the vertices of the object's face that is facing the tray; $(A4)$ point contacts with friction model are used to define the set of wrenches that are transmitted across the contacts; $(A5)$ the friction coefficient, μ, between the object and the tray is uniform and known.

B. Integrated Manipulator-Object Dynamics in Task Space

The dynamic equation of motion for a n-DoF serial link manipulator in the joint space can be written as

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{c}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) = \boldsymbol{\tau} - \boldsymbol{\tau}_{ext}, \tag{1}$$

where $\mathbf{M} \in \mathbb{R}^{n\times n}$, $\mathbf{c} \in \mathbb{R}^n$, $\mathbf{g} \in \mathbb{R}^n$, $\boldsymbol{\tau} \in \mathbb{R}^n$, and $\boldsymbol{\tau}_{ext} \in \mathbb{R}^n$ are the joint space inertia matrix, Coriolis and centrifugal forces, gravitational forces, joint actuation torques, and torque due to external forces acting on the system, respectively. In (1), $\mathbf{q}$, $\dot{\mathbf{q}}, \ddot{\mathbf{q}} \in \mathbb{R}^n$ are the joint angles and their respective derivatives. Equation (1) can be written in task space coordinates defined with respect to the robot's base coordinate $\mathbf{O}_{\text{rob}}$ (spatial) as

$$\bar{\mathbf{M}}(\mathbf{x})\ddot{\mathbf{x}} + \bar{\mathbf{c}}(\mathbf{x},\dot{\mathbf{x}}) + \bar{\mathbf{g}}(\mathbf{x}) = \mathbf{F} - \mathbf{F}_{ext}, \tag{2}$$

where $\bar{\mathbf{M}} = (\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T)^{-1}$, $\bar{\mathbf{c}} = \bar{\mathbf{M}}(\mathbf{J}\mathbf{M}^{-1}\mathbf{c} - \mathbf{J}\dot{\mathbf{q}})$, $\bar{\mathbf{g}} = \bar{\mathbf{M}}(\mathbf{J}\mathbf{M}^{-1}\mathbf{g})$. In (2), $\mathbf{x} \in \mathbb{R}^{n_t}$ is the task coordinate vector, with $n_t > 0$, and $\mathbf{J} \in \mathbb{R}^{n_t\times n}$ is its corresponding task Jacobian. Similar to (2), assuming the object to be a rigid body, the dynamic equation of motion of the object can be written in the object's body coordinate ($\mathbf{O}_{\text{obj}}$) as

$$\mathbf{M}_o(\mathbf{x}_o)\dot{\nu} + \mathbf{C}_o(\mathbf{x}_o,\nu)\nu + \mathbf{g}_o(\mathbf{x}_o) = \mathbf{F}_o, \tag{3}$$

where $\mathbf{M}_o \in \mathbb{R}^{6\times6}$, $\mathbf{C}_o \in \mathbb{R}^{6\times6}$, $\mathbf{g}_o \in \mathbb{R}^6$, and $\mathbf{F}_o \in \mathbb{R}^6$ are the object's inertia matrix, centrifugal and Coriolis matrix, gravity vector, and the wrench acting on the object, respectively. In (3), $\mathbf{x}_o = (\mathbf{p}, \mathbf{o})$ is the object's pose composed by the position $\mathbf{p} \in \mathbb{R}^3$ and the orientation $\mathbf{o} \in \mathbb{R}^3$ (Euler angles) or $\mathbb{R}^4$ (quaternion); $\nu = \left[\mathbf{v}^T, \boldsymbol{\omega}^T\right]^T \in \mathbb{R}^6$ is the object twist, where $\mathbf{v} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$ represent the object's linear and angular velocities, respectively; $\mathbf{F}_o = \left[\mathbf{f}_o^T, \boldsymbol{\tau}_o^T\right]^T \in \mathbb{R}^6$ denotes the object wrench with $\mathbf{f}_o \in \mathbb{R}^3$ and $\boldsymbol{\tau}_o \in \mathbb{R}^3$ representing the force and torque acting on the object, respectively.

For the non-prehensile object transportation scenario, (3) can be included in (2) by considering the object as an external load acting on the robot. This assumption is valid since the object is expected not to slide with the controller presented later in Section III. The above inclusion, however, cannot be done directly since equations (3) and (2) are in $\mathbf{O}_{\text{obj}}$ (body) and $\mathbf{O}_{\text{rob}}$ (spatial) coordinates, respectively. By computing $\mathbf{x}_o$, ν, $\dot{\nu}$ in $\mathbf{O}_{\text{rob}}$ and substituting them in (3), the object's dynamic equation can be obtained as

$$\mathbf{M}_o^s(\mathbf{x}_o^s)\dot{\nu}^s + \mathbf{C}_o^s(\mathbf{x}_o^s,\nu^s)\nu^s + \mathbf{g}_o^s(\mathbf{x}_o^s) = \mathbf{F}_o^s. \tag{4}$$

In (4), $\mathbf{x}_o^s = \mathbf{T}_{\text{o}}^{\text{s}}\mathbf{x}_o$, $\nu^s = \mathbf{A}\mathbf{d}^{-1}\nu$, and $\dot{\nu}^s = \mathbf{A}\mathbf{d}^{-1}\dot{\nu}$, where $\mathbf{T}_{\text{o}}^{\text{s}} \in \mathbb{SE}(3)$ is a coordinate transformation matrix and $\mathbf{Ad}$ denotes the adjoint matrix to transform twists from the body to spatial coordinates [17]. Taking the task point to be the object's center of mass (CoM) in (2) results in $\dot{\nu}^s = \ddot{\mathbf{x}}$ since they are represented in $\mathbf{O}_{\text{rob}}$. Substituting $\mathbf{F}_o^s$ from (4) as $\mathbf{F}_{ext}$ in (2) and using the above similarity condition ($\dot{\nu}^s = \ddot{\mathbf{x}}$) yields

$$(\bar{\mathbf{M}} + \mathbf{M}_o^s)\ddot{\mathbf{x}} + (\bar{\mathbf{c}} + \mathbf{C}_o^s\dot{\mathbf{x}}) + (\bar{\mathbf{g}} + \mathbf{g}_o^s) = \mathbf{F}. \tag{5}$$

The above manipulator-object integrated dynamic equation in the task space can be succinctly written as

$$\bar{\mathbf{M}}_u\ddot{\mathbf{x}} + \bar{\mathbf{c}}_u + \bar{\mathbf{g}}_u = \mathbf{F}. \tag{6}$$

This is used in the rest of the work as the system model to compute the desired control actions.

C. Contact Modeling and Friction Cone Constraints

The contact model and the friction cone constraints used in this work are similar to the ones proposed in [13]. Any $\mathbf{F}_o$ can be achieved through a set of wrenches exerted at the n_c number of contact points located along the periphery of the object's face in contact with the tray. Based on A1, A2, and A3, $n_c = 4$, and its location are known. As per A4, only forces $\mathbf{f}_{c_i} \in \mathbb{R}^3$ are considered to be transmitted through the contact and no torques. The set of all contact forces $\mathbf{F}_c = \left[\mathbf{f}_{c_1}^T, \cdots, \mathbf{f}_{c_{n_c}}^T\right]^T \in \mathbb{R}^{3n_c}$ can be related to the object wrench $\mathbf{F}_o$ in its body coordinates as $\mathbf{F}_o = \mathbf{G}\mathbf{F}_c$, where $\mathbf{G} \in \mathbb{R}^{6\times 3n_c}$ is denoted as *grasp matrix*. For convenience, the friction cone at each contact is approximated by a pyramid. With this approximation, the friction cone consistent $\mathbf{F}_c$ can be represented as $\mathbf{F}_c = \hat{\mathbf{F}}_c\Lambda$, where $\hat{\mathbf{F}}_c \in \mathbb{R}^{3n_c\times kn_c}$ denotes the n_c friction cones and $\Lambda = \left[\lambda_{c_1,1}, \ldots, \lambda_{c_{n_c},k}\right]^{\mathrm{T}} \in \mathbb{R}^{kn_c}$ is a vector of positive scalars used to parameterize contact forces. The aforementioned formulations are used in Section III to optimize the contact forces. For more details regarding $\mathbf{F}_c$, $\mathbf{G}$, $\hat{\mathbf{F}}_c$, Λ, and their relation please refer to [13] and [17].

III. QP Problem Formulation

In this section, the non-prehensile transportation task is formulated as a QP problem by including the various system constraints, such as the system's dynamics, joint position, velocity, and torque limits, and application constraints, such as non-sliding contacts and trajectory tracking. The various models developed in the previous section are actively used here to compute the optimal control actions. The QP problem is written as a minimization problem with various equality and inequality constraints as follows

$$\begin{aligned}
\min_{\dot{\mathbf{x}},\tilde{\ddot{\mathbf{x}}},\mathbf{F},\Lambda,\mathbf{s}} \quad & ||\dot{\mathbf{x}}||_{\mathbf{W}_1} + ||\tilde{\ddot{\mathbf{x}}}||_{\mathbf{W}_2} + ||\mathbf{F}||_{\mathbf{W}_3} + ||\Lambda||_{\mathbf{W}_4} + ||\mathbf{s}||_{\mathbf{W}_5} \\
\text{s.t.} \quad & \bar{\mathbf{M}}_u\ddot{\mathbf{x}} + \bar{\mathbf{c}}_u + \bar{\mathbf{g}}_u = \mathbf{F} \\
& \mathbf{G}^{\dagger}(\mathbf{F}_o) = \hat{\mathbf{F}}_c\Lambda \\
& \Lambda \geq 0 \\
& \ddot{\mathbf{x}}_l \leq \ddot{\mathbf{x}} \leq \ddot{\mathbf{x}}_u \\
& \dot{\mathbf{x}} = \dot{\mathbf{x}}_0 + \ddot{\mathbf{x}}\delta t \\
& \boldsymbol{\tau}_l \leq \mathbf{J}^T\mathbf{F} \leq \boldsymbol{\tau}_u \\
& \dot{\mathbf{q}}_l \leq \mathbf{J}^{\dagger}\dot{\mathbf{x}} \leq \dot{\mathbf{q}}_u \\
& \mathbf{q}_l \leq \mathbf{q}_0 + (\mathbf{J}^{\dagger}\dot{\mathbf{x}})\delta t \leq \mathbf{q}_u,
\end{aligned} \tag{7}$$

where, $\dot{\mathbf{x}} = \left[\mathbf{v}^T, \boldsymbol{\omega}^T\right]^T$ is the Cartesian velocity of the task point in spatial coordinates and its derivative is represented by $\ddot{\mathbf{x}}$. The task acceleration error is $\tilde{\ddot{\mathbf{x}}} = \ddot{\mathbf{x}} - \ddot{\mathbf{x}}^d$, where $\ddot{\mathbf{x}}^d$ represents the desired acceleration trajectory. In the above objective function, $\mathbf{F}$ is the generalized task force from (6), $\mathbf{s}$ is a vector of slack variables that will be discussed later, and $\mathbf{W}_i$ represent the different weights assigned for various objectives, for instance, $||\mathbf{x}||_{\mathbf{W}_1} = \mathbf{x}^T\mathbf{W}_1\mathbf{x}$. In the objective function, $||\tilde{\ddot{\mathbf{x}}}||$ ensures the tracking of the desired trajectory, $||\mathbf{F}||$ reduces the effort required, $||\Lambda||$ minimizes the magnitude of a positive scalar vector that restricts the object slip, and $||\dot{\mathbf{x}}||$ minimizes the task associated twist.

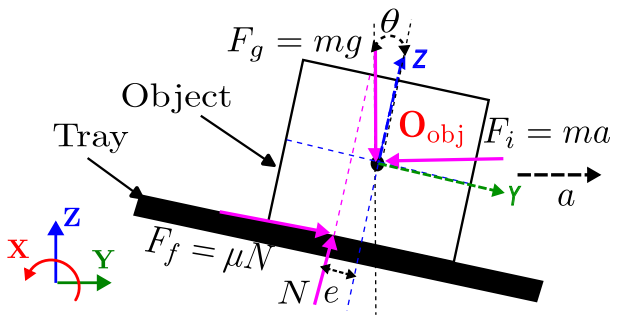

Fig. 2: A tray with an object resting on top of it is shown here. The tray is accelerating along the Y direction and the various forces acting on the object are shown in the object's body coordinate $\mathbf{O}^{\text{b}}_{\text{obj}}$.

The optimal values for $\dot{\mathbf{x}}$, $\tilde{\ddot{\mathbf{x}}}$, $\mathbf{F}$, Λ, and $\mathbf{s}$ are determined by subjecting the objective function to various constraints and the optimal values are denoted by the apex *. The first constraint ensures that the integrated dynamics of the system is consistent; the second one maintains the relation between $\mathbf{F}_c$ and $\mathbf{F}_o$ by substituting $\mathbf{F}_c = \hat{\mathbf{F}}_c\Lambda$ in $\mathbf{F}_o = \mathbf{G}\mathbf{F}_c$, where $\mathbf{F}_o$ can be computed from (4) as $\mathbf{F}_o = \mathbf{A}\mathbf{d}^{-1}\mathbf{F}^s_o$; the third one is aimed at maintaining $\mathbf{F}_c$ within the friction cone to avoid the object slipping during its transportation. This is followed by a limit constraint on the task acceleration to avoid arbitrary acceleration, and an equality constraint is set on $\dot{\mathbf{x}}$ to make sure that the task velocity components are in accordance with those of $\ddot{\mathbf{x}}$. The desired motion is also subjected to several joint-level constraints such as position, velocity, and torque to ensure its physical realization in real systems. The Jacobian $\mathbf{J}$ considered for the last three constraints is nothing but the task Jacobian discussed in Sec. II-B. Lastly, $\mathbf{s} \in \mathbb{R}^{36}$ represents the list of slack variables associated with the integrated dynamics equality constraint (6), lambda inequality constraint $(4 \times n_c)$, joint torque τ (n), and joint velocity (n) limit constraints. The inclusion of $\mathbf{s}$ makes the constraints relatively soft and thereby ensures that the solver always finds a solution, even in extreme situations, without getting into an infeasibility mode. The constraint boundaries' relative softness is tuned by appropriately choosing $\mathbf{W}_5$.

IV. Adaptive Tilting Function

In the previous section, the desired motion of the object is given in the form of the desired acceleration $\ddot{\mathbf{x}}^d \in \mathbb{R}^6$. Ideally, the angular components of $\ddot{\mathbf{x}}^d$ will be zero for a translational motion. However, for very high acceleration/deceleration demanding motions, non-zero rotational motion is necessary to minimize the object's slippage, as explained here using D'Alembert's principle.

Let us consider an object of mass m resting on a tray, as shown in Fig. 2, accelerating at a m/s^2 along the positive Y axis. The object will be subjected to a gravitational force of $F_g = mg$ and an inertial force of $F_i = ma$. Since the tray, along

with the object, is accelerating in the positive Y direction, F_i will be acting in the opposite direction, i.e., negative Y direction, to resist the object's motion. In addition, since the object is resting on the tray, a normal resisting force (resultant) N will be acting to resist F_g through the contacts at an offset e from the center. The friction force resists the object's motion with respect to the tray $F_f = \mu N$ acting tangential to the contact. The object-tray motion shown in Fig. 2 is dynamic, however, the system's equilibrium can be analyzed in static form by applying D'Alembert's principle. Due to the assumption $(A1)$, the object is more likely to slip than to undergo a tilt and hence the conservation of angular momentum is not considered here. Following this, the various forces acting on the object can be resolved along the Z and Y axes in $\mathbf{O}_{\text{obj}}$ as

$$\begin{aligned} \sum F_z &\rightarrow N - mg\cos(\theta) - ma\sin(\theta) = 0, \\ \sum F_y &\rightarrow mg\sin(\theta) + \mu N - ma\cos(\theta) = 0. \end{aligned} \tag{8}$$

For the system considered in Fig. 2, the object can slide along the tray on the Y axis since it is being transported in a non-prehensile manner. To arrest this sliding, dynamic equilibrium needs to be maintained along F_y shown in (8). This can be done in three possible ways: (i) increasing the contact friction, (ii) reducing the acceleration a, and (iii) manipulating the tray orientation θ. While option (i) cannot be done instantaneously, (ii) can severely affect the trajectory tracking performance, and hence, option (iii) is chosen. Accordingly, in (8), by substituting $N = mg\cos(\theta) + ma\sin(\theta)$ from F_z in F_y, the resulting forces along Y axis is

$$mg\sin(\theta) + \mu(mg\cos(\theta) + ma\sin(\theta)) - ma\cos(\theta) = 0. \tag{9}$$

Dividing (9) by $m\cos(\theta)$ and simplifying further yields

$$\tan(\theta) = \frac{\mu g - a}{g + \mu a} \rightarrow \theta = \tan^{-1}\left(\frac{\mu g - a}{g + \mu a}\right). \tag{10}$$

Solving the above equation at every instant can determine the tray orientation that needs to be maintained to avoid object slippage for some given μ and a. For accelerations along the Y axis, θ will be about the axis perpendicular to Y and Z, i.e., the X axis. Similarly, θ will be around the Y-axis for accelerations along the X-axis.

V. OSC Framework

In this section, the control framework that is used to realize the optimal non-prehensile motion is briefly explained. The QP problem introduced in Sec. III, which determines $\mathbf{F}^\star$, and the adaptive tilting function discussed in Sec. IV, which computes θ that can compensate for high accelerations with minimal object slippage, are integrated here. The optimal motion is realized using a multi-tasking OSC.

Given an object's goal position, $\mathbf{x}_g$, along with their respective timeline, $\mathbf{t}_g$, a quintic polynomial trajectory ($\mathbf{x}_p^d$, $\dot{\mathbf{x}}_p^d$, $\ddot{\mathbf{x}}_p^d$) is generated with the initial and end velocities and accelerations set to 0. The generated trajectory for the translational motion is given as an input to the adaptive tilting function block. Here, the maximum acceleration/deceleration is determined from $\ddot{\mathbf{x}}_p^d$, and θ_g along X and Y axes are analytically computed, as

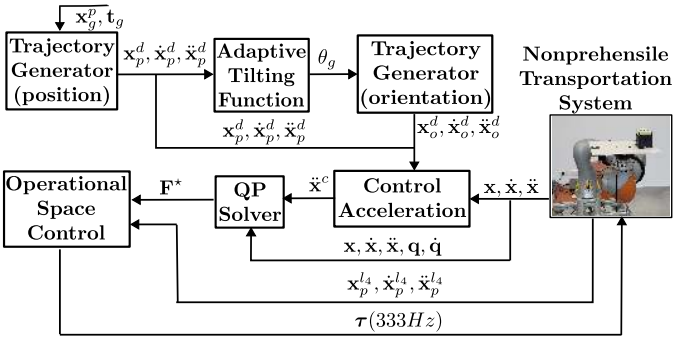

Fig. 3: The OSC framework to realize the non-prehensile motion is shown.

explained in Section IV. Using θ_g and $\mathbf{t}_g$, quintic orientation trajectory ($\mathbf{x}_o^d$, $\dot{\mathbf{x}}_o^d$, $\ddot{\mathbf{x}}_o^d$) is planned with quaternion polynomials as described in [18]. The generated position and orientation trajectories are used to compute the control acceleration $\ddot{\mathbf{x}}^c \in \mathbb{R}^{6\times 1}$ as

$$\ddot{\mathbf{x}}^c(t) = \begin{bmatrix} \ddot{\mathbf{x}}_p^d(t) \\ \ddot{\mathbf{x}}_o^d(t) \end{bmatrix} + \mathbf{K}_p \begin{bmatrix} \mathbf{e}_p(t) \\ \mathbf{e}_o(t) \end{bmatrix} + \mathbf{K}_d \begin{bmatrix} \dot{\mathbf{e}}_p(t) \\ \dot{\mathbf{e}}_o(t) \end{bmatrix}, \tag{11}$$

where, $\mathbf{e}_p(t) = \mathbf{x}_p^d(t) - \mathbf{x}_p$ and $\mathbf{e}_o(t) = \mathbf{x}_o^d(t) - \mathbf{x}_o \in \mathbb{R}^{3\times 1}$ are the position and orientation errors, respectively, and their derivatives are represented as $\dot{\mathbf{e}}_p(t)$ and $\dot{\mathbf{e}}_o(t)$. The QP solver module gets $\ddot{\mathbf{x}}^c$, the object's instantaneous pose ($\mathbf{x}$) and its derivatives, $\mathbf{q}$, and $\dot{\mathbf{q}}$ as input. Here, the QP problem (7) is solved to determine the optimal task space force $\mathbf{F}^\star$.

The non-prehensile transportation of the object is realized through a multitasking OSC. Two tasks are considered here for a successful and stable motion realization: 1) non-prehensile object transportation task, and 2) robot posture maintenance task. Of the two tasks, the first one is the primary task, and the second one is executed within the null space of the primary. The total OSC torque is written as

$$\boldsymbol{\tau} = \boldsymbol{\tau}_{\text{npt}} + \boldsymbol{\tau}_{\text{pos}}, \tag{12}$$

where $\boldsymbol{\tau}_{\text{npt}} \in \mathbb{R}^n$ and $\boldsymbol{\tau}_{\text{pos}} \in \mathbb{R}^n$ is the torque corresponding to the non-prehensile transportation and posture maintenance task, respectively. Accordingly, the dynamically consistent torque associated with the non-prehensile transportation task is computed as

$$\boldsymbol{\tau}_{\text{npt}} = \mathbf{J}_{\text{npt}}^{\text{T}} \mathbf{F}_{\text{npt}}, \tag{13}$$

where $\mathbf{J}_{\text{npt}}^{\text{T}} \in \mathbb{R}^{6\times n}$ is the jacobian of the object's CoM in spatial coordinates and $\mathbf{F}_{\text{npt}} \in \mathbb{R}^6$ is the non-prehensile transportation task force. By substituting $\mathbf{F}^\star$ from (7) for $\mathbf{F}_{\text{npt}}$ in (13), the optimal non-prehensile transportation task can be realized.

The secondary task (posture task) is intended to maintain a favorable posture of the robot during the complete non-prehensile transportation task. For instance, in our case, the robot configuration shown in Fig. 1 can result in link 4 touching the ground, and this can have some disturbing effect on the non-prehensile transportation task. This is addressed by maintaining the z position of link 4 CoM in its initial position. The torque associated with this task can be computed as

$$\boldsymbol{\tau}_{\text{pos}} = \mathbf{J}_{\text{pos}}^{\text{T}} F_{\text{pos}}, \tag{14}$$

$$F_{\text{pos}} = \bar{M}_{\text{pos}} \ddot{x}_z^r + \bar{c}_{\text{pos}} + \bar{g}_{\text{pos}}, \tag{15}$$

where, $\mathbf{J}_{\text{pos}} \in \mathbb{R}^{1\times n}$ is the Jacobian of link 4's CoM and $F_{\text{pos}} \in \mathbb{R}^1$ is the posture task force and it is computed as in (15). $\bar{M}_{\text{pos}} \in \mathbb{R}^1$, $\bar{c}_{\text{pos}} \in \mathbb{R}^1$, and $\bar{g}_{\text{pos}} \in \mathbb{R}^1$ denote the inertial matrix, Coriolis, and gravity force vector associated with the posture task and they are computed similar to (2). The required acceleration of link 4 CoM $\ddot{x}_z^r$ is computed as

$$\ddot{x}_z^r = \ddot{x}_z^d + k_d^e(\dot{x}_z^d - \dot{x}_z) + k_p^e(x_z^d - x_z) \tag{16}$$

where $\ddot{x}_z^d$, $\dot{x}_z^d$, and x_z^d represent the desired position, velocity, and acceleration of link 4 CoM's z coordinate and its corresponding instantaneous values are denoted by $\ddot{x}_z$, $\dot{x}_z$, and x_z respectively. The proportional and derivative gains are denoted by k_p^e and k_d^e, respectively.

The posture control torque that does not interfere with the non-prehensile transportation task can be determined by projecting $\boldsymbol{\tau}_{\text{pos}}$ into the null space of $\boldsymbol{\tau}_{\text{npt}}$, as $\tau_{\text{pos|npt}} = \mathbf{N}_{\text{npt}}^{\text{T}}\boldsymbol{\tau}_{\text{pos}}$, where $\mathbf{N}_{\text{npt}} = \mathbf{I} - \mathbf{J}_{\text{npt}}^{\text{T}}\bar{\mathbf{J}}_{\text{npt}}^{\text{T}}$ is the null space of the transportation task, in which $\mathbf{I} \in \mathbb{R}^{7\times 7}$ is an identity matrix and $\bar{\mathbf{J}}_{\text{npt}} = \bar{\mathbf{M}}_u^{-1}\mathbf{J}_{\text{npt}}^{\text{T}}(\mathbf{J}_{\text{npt}}\bar{\mathbf{M}}_u^{-1}\mathbf{J}_{\text{npt}}^{\text{T}})^{-1}$. The total OSC torque $\boldsymbol{\tau}$ is computed by replacing $\boldsymbol{\tau}_{\text{pos}}$ with $\boldsymbol{\tau}_{\text{pos|npt}}$ in (12).

VI. Simulation and Results

This section reports the various numerical evaluations performed and their corresponding results. The simulation setup involves a 7 DoF KUKA LBR iiwa 7 R800 robot with a square plate (0.35×0.35×2.5e-3 [m], m_t = 0.15 [kg], I_t = diag([1.5e-3, 1.5e-3, 3e-3]) [kgm^2]) mounted on its end-effector acting as a tray, and the object that is transported is a cuboid (0.04 [m], m_o = 0.1 [kg], μ = 0.35, and I_o = diag(1e-4) [kgm^2]). The transported object's pose and its derivatives with respect to the global frame are taken from the simulator. The complete set of simulations is carried out using the Gazebo simulator. For solving the QP problem (7), the OSQP solver [19] has been used.

A. Numerical Evaluations

The proposed controller is subjected to several numerical evaluations to understand and ascertain its performance. The first set of evaluations analyses the effect of including or excluding the system constraints with those of the application ones. Secondly, the contribution of the adaptive tilting function (ATF) is quantified in different scenarios. Finally, the performance of the proposed controller is compared to other existing controllers, and their results are critically analyzed. A linear motion covering 0.6 m in 1 s along the positive Y axis is considered for the first analysis. For the last two, a much faster motion is taken by reducing the time to 0.85 s, highlighting the proposed controller's agility and slippage reduction. In the rest of the section, the above motions are called *test motion 1* and *test motion 2*, respectively. The distance for the above motions is chosen considering the robot's reach, and the time is low enough to generate high acceleration that can result in the object's slippage.

B. With and Without System Constraints

In (7), the first five constraints represent the application constraints, and the last three denote the system constraints. Considering all the above constraints, *test motion 1* is evaluated in the simulator following the control framework shown in Fig. 3. The results obtained are shown in Fig. 4. Since the linear motion is only along the Y axis, the ATF generates orientation motion only along the X axis to counter object slippage in the Y direction. For brevity, only relevant results are shown in Fig. 4. The desired position trajectory ($\mathbf{p}_y^d$) and the resulting optimal motion ($\mathbf{p}_y^\star$) of the object along the Y axis is shown in the top left corner of Fig. 4a and its respective orientation ($\mathbf{o}_x^d$, $\mathbf{o}_x^\star$) to minimize the object slippage is shown in the top right corner. The desired and optimal force ($\mathbf{f}_y^d$, $\mathbf{f}_y^\star$) acting on the object in the Y direction and its respective torque (τ_x^d, $\tau_x^\star$) acting along the X direction are shown in the first column of Fig. 4b. On the right side of Fig. 4b, $\mathbf{f}_y^\star$ is shown at the top, and the resulting object slip ($\|slip\|$) is plotted at the bottom.

In the acceleration plot of Fig. 4a, $\ddot{\mathbf{p}}_y^c$ is computed as given in (11), and it is observed to be closely followed by $\ddot{\mathbf{p}}_y^\star$, except for a few instances. This discrepancy is due to the QP solver that tries to achieve both the trajectory tracking task and the slip minimization task. In the case of $\ddot{\mathbf{o}}_x$, since the tilting is done to counter the object slippage, $\ddot{\mathbf{o}}_x^\star$ follows $\ddot{\mathbf{o}}_x^c$ closely and $\ddot{\mathbf{o}}_x^c > \ddot{\mathbf{o}}_x^d$ because of the feedback gains. A similar trend is observed in the force and torque plots of Fig. 4b. $\mathbf{f}_y^d$ and τ_x^d are computed using the control accelerations $\ddot{\mathbf{p}}_y^c$ and $\ddot{\mathbf{o}}_x^c$, and their values are plotted here in $\mathbf{O}_{\text{obj}}$. The position plot in Fig. 4a (top left) shows the perfect tracking of its desired trajectory. In the orientation plot (top right), since the object is accelerating in the positive Y direction from $0-0.25$ s, as seen in Fig. 4a(bottom left), the object is oriented in the negative direction. This is done to resist the slipping by directing a part of F_i to act in the normal direction, which contributes to the increase of F_f, as shown in Fig. 2. This, in turn, maintains the dynamic equilibrium of the object. Vice-versa action is observed during the acceleration of the object in the negative Y direction around $0.6-0.75$ s. The discrepancy observed between $\mathbf{o}_x^\star$ and $\mathbf{o}_x^d$ is due to the controller that tries to optimize $\mathbf{o}_x$ to minimize the object slippage. This is also reflected in the $\ddot{\mathbf{o}}_x$ plot.

Regarding the object slip, a maximum of $\approx$ 6mm is observed and it occurs just after the onset of $\mathbf{f}_y^\star \approx 0.4$N around 0.25s, as seen in the top right plot of Fig. 4b. It is to be noted that $\mathbf{f}_y^\star(0.4\text{N}) > F_f(0.34\text{N})$ (taking $\mu = 0.35$ and $N = 0.1\cdot 9.8$) and can certainly result in the object sliding. However, the proposed controller minimizes the sliding significantly and realizes the desired motion.

Without System Constraints: To understand the significance of including the system constraints in addition to those of the application ones, *test motion 1* is repeated without the system constraints, and the results are compared in Fig. 5. The results obtained with and without the system constraints are denoted by wSC and woSC, respectively. Though (7) includes the joint positions, velocities, and torques as system constraints, only the violated variables are reported in Fig. 5 for brevity. The

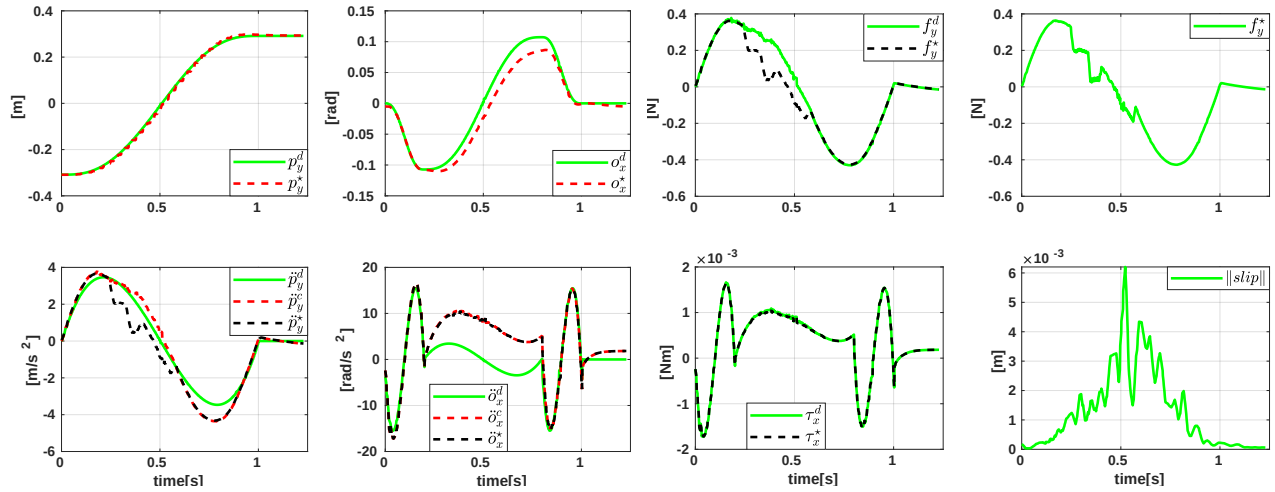

Fig. 4: (a) The desired ($\mathbf{p}_y^d$) and optimal ($\mathbf{p}_y^\star$) trajectory of the object along the Y axis is shown in the top left corner. The top right corner plot shows the desired ($\mathbf{o}_x^d$) and optimal ($\mathbf{o}_x^\star$) orientation trajectory along the X axis. The bottom plots shows the desired ($\ddot{\mathbf{p}}_y^d$, $\ddot{\mathbf{o}}_x^d$), commanded ($\ddot{\mathbf{p}}_y^c$, $\ddot{\mathbf{o}}_x^c$), and optimal acceleration ($\ddot{\mathbf{p}}_y^\star$, $\ddot{\mathbf{o}}_x^\star$) of $\mathbf{p}_y^d$ and $\mathbf{o}_x^d$. (b) The desired and optimal force along Y axis ($\mathbf{f}_y^d$, $\mathbf{f}_y^\star$) and torque along X axis (τ_x^d, $\tau_x^\star$) is shown in the first column respectively. The second column shows $\mathbf{f}_y^\star$ at the top and the resulting object slip norm ($\|slip\|$) at the bottom plot.

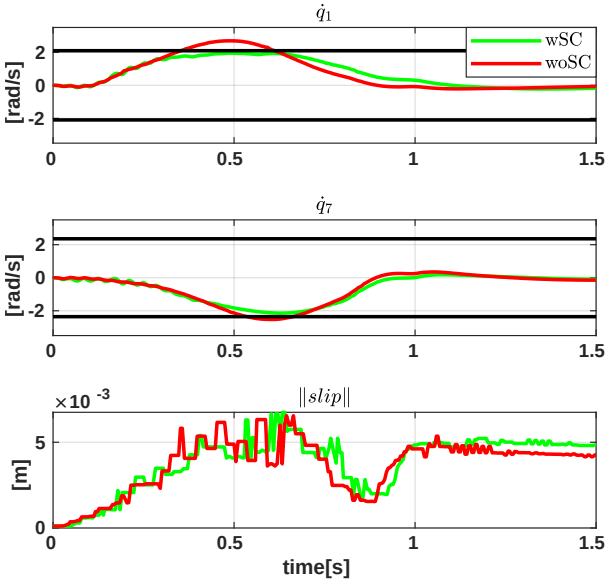

Fig. 5: The manipulator's joint velocities (first two graphs) that violate their limits and the object slip (bottom) are compared between two different simulations: with system constraints (wSC) and without system constraints(woSC).

first two plots in Fig. 5 compare the joint velocities of joints 1 and 7, respectively. As seen in the figure, in the case of woSC, $\dot{q}_1$ and $\dot{q}_7$ violate their respective limits around 0.5s and 0.6s, respectively. Whereas, with wSC, $\dot{q}_1$ and $\dot{q}_7$ are maintained within their respective limits. The last plot compares $\|slip\|$, and we can observe that the maximum slip is almost the same. This suggests that wSC maintains the same level of performance despite considering the system constraints and hence, ensures the successful realization of *test motion 1* with a real system.

C. Significance of Adaptive Tilting Function

The contribution of the proposed ATF is ascertained by executing *test motion 2* with two different settings: 1) with ATF (wAT) and 2) without ATF (woAT). The results obtained are plotted in Fig. 6. With woAT, the controller tries to minimize $\|slip\|$ by reducing the force acting on the object in the Y direction ($\mathbf{f}_y^\star$), as seen in the bottom left plot. However, since the controller needs to follow the desired trajectory as well, the slip reduced by optimizing the $\mathbf{f}_y^\star$ is minimal, as observed in the $\|slip\|$ plot (bottom right). A maximum slip of ≈ 26 mm is achieved with woAT. In comparison, with wAT, the proposed controller can reduce the object's slip significantly ($< 5mm$) using ATF and, at the same time, maintain high $\mathbf{f}_y^\star$ to track the desired trajectory. This can be seen in the plots of $\mathbf{f}_y^\star$ and $\mathbf{p}_y$ respectively. The reduction in $\mathbf{f}_y^\star$ and relatively high slip affect the tracking of woAT, as visualized in the $\mathbf{p}_y$ plot. This shows a mediocre performance of woAT in minimizing object slippage and trajectory tracking. Whereas, wAT minimizes both the object sliding and also delivers better tracking of the desired trajectory. The main reason for the performance difference is that wAT uses the orientation to handle $\|slip\|$ employing ATF and $\mathbf{f}_y^\star$ to track the desired trajectory. In the case of woAT, without ATF, the controller is forced to use $\mathbf{f}_y^\star$ to track the desired trajectory and to minimize $\|slip\|$, hence, the mediocre results.

D. Comparison of Different Controllers

The performance of the proposed controller is compared to two contemporary controllers: 1) classical inverse dynamics (ID), as given in (1) with $\tau_{ext} = 0$; and 2) operational space controller with object dynamics (OSC) stated in (6). The joint references for ID are computed with $\ddot{\mathbf{x}}^d, \dot{\mathbf{x}}^d$, and $\mathbf{x}^d$ using inverse kinematics. The proposed non-prehensile object transportation controller is denoted as NoT. For comparison, $\mathbf{p}_y$ and $\|slip\|$ obtained with *test motion 2* are considered, as shown in Fig. 7.

Observing the $\mathbf{p}_y$ plot, it can be seen that the tracking of NoT is much better than the other two controllers. Between OSC and ID, the performance of the former is slightly better. In the case of $\|slip\|$, NoT minimizes $\|slip\|$ to a considerable

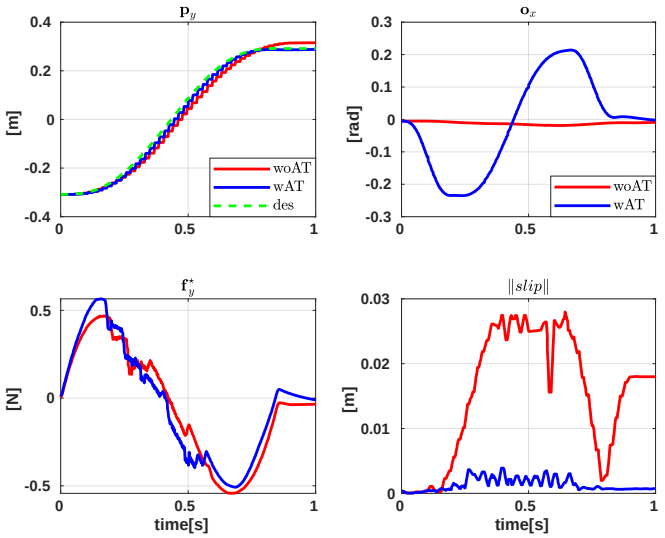

Fig. 6: The significance of adaptive tilting function (ATF) is evaluated by comparing two sets of results: with ATF(wAT) and without ATF(woAT). The Y-direction position trajectory, X-direction orientation, the force acting on the object in the Y direction (body coordinates), and ||slip|| are compared.

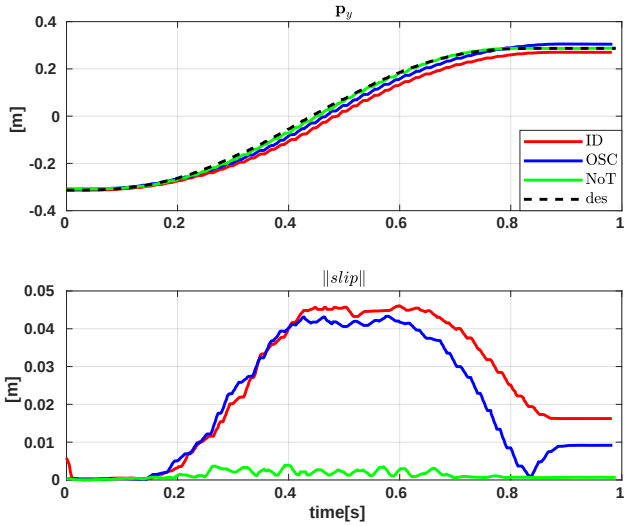

Fig. 7: Three different types of controllers are compared: 1) ID, 2) OSC, and 3) NoT. The plot compares p_y (top) and $||slip||$ (bottom).

extent by optimizing its control actions and thus maintaining a slip of $<$ 5mm throughout the motion. This is not the case with the other two controllers, which only track the desired trajectory. As a result, a maximum of $||slip|| \approx 45$ mm is observed with both controllers. It is to be noted that $||slip||$ could have been even greater with ID and OSC; however, it is limited due to the quintic polynomial trajectory used as a reference. In addition, no significant difference is observed between ID and OSC despite the latter considering the object dynamics explicitly. This could be due to the dynamic properties of the object itself, which is too small to have any considerable effect. However, OSC reports a slightly better performance when compared to ID. Overall, the proposed controller (NoT) reports better tracking and object slip minimization performance.

TABLE I: Dynamic properties and control parameters

Tray	plate (0.30×0.15×0.05 m), m = 0.2 [Kg], and $\mathbf{I}_t$ = diag([1.5e-3, 1.5e-3, 3e-3]) [Kgm^2]
Transported object	cuboid (0.065 [m]), m = 0.366 [Kg], μ = 0.25, and $\mathbf{I}_o$ = diag([3.49e-4, 3.49e-4, 3.96e-4]) [Kgm^2]
Control parameters	$\mathbf{K}_p$ = diag([35,35,35,380,380,380]), $\mathbf{K}_d = \sqrt{\mathbf{K}_p}\times 2$, k_p^e = 5.0, and $k_d^e = \sqrt{k_p^e}$ $\mathbf{W}_1 = \mathbf{I}_{6\times6}$, $\mathbf{W}_2$ = diag([100,100,325,250,250,250]), $\mathbf{W}_3 = \mathbf{I}_{6\times6} \times 0.025$, $\mathbf{W}_4 = \mathbf{I}_{6\times6} \times 2e4$ $\mathbf{W}_5$ = diag([$\mathbf{I}_{4\times n_c\times4\times n_c} \times 5e4$, $\mathbf{I}_{n\times n} \times 200$, $\mathbf{I}_{n\times n} \times 100$, $\mathbf{I}_{6\times6} \times 100$, $\mathbf{I}_{3\times n_c\times3\times n_c}$]).

VII. Experiments and Results

In this section, the various experiments carried out to demonstrate the proposed controller's performance are reported, along with their respective results. The experimental setup includes an LBR iiwa 7 R800 manipulator, a 3D printed plastic (ABS) plate that acts as the transportation tray, and a hollow mild steel cube with a plastic base taken as the object to be transported, as shown in Fig. 1. An April tag is attached to both the object and the tray to monitor the relative slipping of the former with respect to the latter during transportation. A camera is mounted at the top to record the complete motion of the system at 240 frame per second. This video is later processed offline to extract each tag's respective pose using [20] and subsequently, the object's slip. For this experiment, the manipulator is torque controlled using the fast robot interface (FRI) [21] at $\approx$ 333 Hz using a quad-core Intel i7 laptop running on Ubuntu 20.04. The parameters used for this experiment are tabulated in Table I*.

For the experimental evaluation, three different trajectories are considered: 1) a trajectory involving a motion of 0.5 m along the Y direction in 1.1 s (T1), 2) a sequence of trajectories involving motion along X and Y directions (T2), and 3) an arc motion (T3). Only T1 results are reported and extensively analyzed in this manuscript, for brevity. For the results of T2 and T3, the readers are directed to the attached video. The desired trajectory for T1 (black dash lines) is shown in the top left of Fig. 8. Similar to the numerical evaluations, the three different controllers are used to execute the desired non-prehensile transportation motion. The results are shown in Fig. 8, in particular, the position tracking along the Y axis ($\mathbf{p}_y$), orientation along the X axis ($\mathbf{o}_x$), linear acceleration along the Y axis ($\ddot{\mathbf{p}}_y$) and $||slip||$ are compared.

The trajectory tracking of all three controllers seems to perform well in tracking the desired trajectory, as shown in the $\mathbf{p}_y$ plot. However, the tracking accuracy of NoT is slightly lower towards the end of the trajectory when compared to the other two controllers. This could be because a very high angular acceleration and deceleration along the X axis is demanded, as seen in the $\mathbf{o}_x$ plot to minimize $||slip||$. This, in turn, demands a faster system response which is quite demanding to achieve with a controller running at 333 Hz on a non-real-time Linux system. However, we can see that the tracking error reduces gradually as time increases. In such demanding situations, depending on the objective weights $\mathbf{W}_i$

*For $\mathbf{W}_i$ in (7), higher weights are considered for $\mathbf{W}_2$ and $\mathbf{W}_4$ to track the desired trajectory and minimize the slip, respectively. Similarly, for $\mathbf{W}_5$, higher weights are assigned for the slack variables related to Λ, $\mathbf{q}$, $\dot{\mathbf{q}}$, and $\mathbf{F}$ to respect the slip, system, and motion constraints respectively.

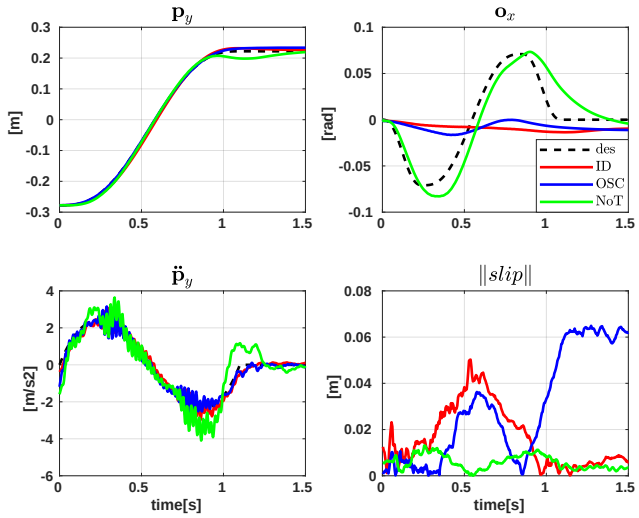

Fig. 8: The experimental results obtained for trajectory T1 with the three different controllers are compared here. The position tracking ($\mathbf{p}_y$) and its acceleration $\ddot{\mathbf{p}}_y$ are shown in the left-side plots and the X orientation ($\mathbf{o}_x$) and the object slip ($\|slip\|$) are shown in the right-side plots.

in (7), the NoT controller tries to minimize the object slip as much as possible by compromising on the trajectory tracking accuracy. The $\mathbf{o}_x$ plot suggests that the tracking accuracy is relatively lower than that of the position tracking. This could also be due to the aforementioned reasons. Besides, the behavior also indicates that the friction model considered for the last three joints may not be sufficient. For this experiment, a simple linear friction model is used for all n joints of the robot.

The ($\ddot{\mathbf{p}}_y$) plot shows that for all three controllers, the resulting peak acceleration during the motion is $> \pm 3\text{m/s}^2$. From the dynamic properties in Table I, one can determine that the motion results in an inertial force $F_i > 1.098$ N, which is considerably greater than $F_f = 0.8967$ N. Hence, T1 can undoubtedly result in the slipping of the object. This is indeed observed in the $\|slip\|$ plot for the ID (0.057 m) and OSC (0.068 m) controllers. However, the proposed NoT controller minimizes $||slip||$ considerably (0.01 m).

VIII. Conclusion

In this paper, we have proposed a dynamic non-prehensile object transportation controller for robot manipulators. The transportation task is devised as a QP problem using the integrated manipulator-object dynamic model, a linear approximation of the friction cone, and the various system's and application's limits as equality and inequality constraints. To enable faster motions without object slipping, an adaptive tilting function is proposed that helps to realize the transportation task reactively. The proposed controller (NoT) has been extensively analyzed numerically. Its performance is better than the conventional ID and OSC controllers, especially in minimizing the object slip during high acceleration motions. The performance of NoT is also experimentally validated and verified with an LBR iiwa 7 manipulator.

In the future, we would like to implement the proposed controller on a real-time system with a higher torque control rate and extend its application to handle arbitrary trajectories by actively considering the singularity and manipulability of any given manipulator. We would also like to consider other scenarios, such as objects prone to tilting, i.e., tall objects with a small footprint, handling multiple objects, and so on.

References

[1] M. T. Mason and K. M. Lynch, "Dynamic manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, vol. 1, 1993, pp. 152–159.

[2] F. Ruggiero, V. Lippiello, and B. Siciliano, "Nonprehensile dynamic manipulation: A survey," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1711–1718, July 2018.

[3] J. Z. Woodruff and K. M. Lynch, "Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 4066–4073.

[4] K.-T. Yu, M. Bauza, N. Fazeli, and A. Rodriguez, "More than a million ways to be pushed. a high-fidelity experimental dataset of planar pushing," in *Proc. IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 2016, pp. 30–37.

[5] A. Zeng, S. Song, J. Lee, A. Rodriguez, and T. Funkhouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *IEEE Trans. Robot.*, vol. 36, no. 4, pp. 1307–1319, 2020.

[6] A. Satici, F. Ruggiero, V. Lippiello, and B. Siciliano, "Coordinate-free framework for robotic pizza tossing and catching," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 3932–3939.

[7] M. M. Schill, F. Gruber, and M. Buss, "Quasi-direct nonprehensile catching with uncertain object states," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2015, pp. 2468–2474.

[8] 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.

[9] P. Lertkultanon and Q.-C. Pham, "Dynamic non-prehensile object transportation," in *Proc. Int. Conf. Control Autom. Robot. Vis.*, 2014, pp. 1392–1397.

[10] H. Gattringer, A. Mueller, M. Oberherber, and D. Kaserer, "Time-optimal robotic manipulation on a predefined path of loosely placed objects: Modeling and experiment," *Mechatronics*, vol. 84, p. 102753, 2022.

[11] K. Lynch and M. Mason, "Dynamic underactuated nonprehensile manipulation," in *Proc. IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, vol. 2, 1996, pp. 889–896 vol.2.

[12] G. Martucci, J. Bimbo, D. Prattichizzo, and M. Malvezzi, "Maintaining stable grasps during highly dynamic robot trajectories," in *Proc. IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 2020, pp. 9198–9204.

[13] M. Selvaggio, J. Cacace, C. Pacchierotti, F. Ruggiero, and P. R. Giordano, "A shared-control teleoperation architecture for nonprehensile object transportation," *IEEE Trans. Robot.*, vol. 38, no. 1, pp. 569–583, 2022.

[14] J. 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 *Proc. IEEE-RAS Int. Conf. Human. Rob.*, 2018, pp. 704–709.

[15] V. Morlando, M. Selvaggio, and F. Ruggiero, "Nonprehensile object transportation with a legged manipulator," in *Proc. Int. Conf. Robot. Autom.*, 2022, pp. 6628–6634.

[16] G. Zhang, S. Ma, Y. Shen, and Y. Li, "A motion planning approach for nonprehensile manipulation and locomotion tasks of a legged robot," *IEEE Trans. Robot.*, vol. 36, no. 3, pp. 855–874, 2020.

[17] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation, Chapter 5*. CRC press, 2017.

[18] M. Shahbazi, N. Kashiri, D. Caldwell, and N. Tsagarakis, "Orientation planning in task space using quaternion polynomials," in *IEEE Int. Conf. Robot. and Biomim.* IEEE, 2017, pp. 2343–2348.

[19] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Math. Program. Comput.*, vol. 12, no. 4, pp. 637–672, 2020.

[20] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3400–3407.

[21] G. Schreiber, A. Stemmer, and R. Bischoff, "The fast research interface for the kuka lightweight robot," in *IEEE ICRA 2010 workshop on innovative robot control architectures for demanding (Research) applications how to modify and enhance commercial controllers*, 2010, pp. 15–21.