

A General Formulation for Path Constrained Time-Optimized Trajectory Planning with Environmental and Object Contacts

Dasharadhan Mahalingam¹, Aditya Patankar¹, Riddhiman Laha², Srinivasan Lakshminarayanan²,
 Sami Haddadin², and Nilanjan Chakraborty¹

Abstract—A typical manipulation task consists of a manipulator equipped with a gripper to grasp and move an object with constraints on the motion of the hand-held object, which may be due to the nature of the task itself or from object-environment contacts. In this paper, we study the problem of computing joint torques and grasping forces for time-optimal motion of an object, while ensuring that the grasp is not lost and any constraints on the motion of the object, either due to dynamics, environment contact, or no-slip requirements, are also satisfied. We present a second-order cone program (SOCP) formulation of the time-optimal trajectory planning problem that considers nonlinear friction cone constraints at the hand-object and object-environment contacts. Since SOCPs are convex optimization problems that can be solved optimally in polynomial time using interior point methods, we can solve the trajectory optimization problem efficiently. We present simulation results on three examples, including a non-prehensile manipulation task, which shows the generality and effectiveness of our approach.

I. INTRODUCTION

Time-optimized motion planning promises attractive new applications that will transform industrial productivity as well as improve service robotics utilization [1]. For example, an optimized motion planner could deliver fragile objects on a factory floor within a stipulated time such that the objects do not fall over. In a service robotics setting, the same planner can be employed to transport a bowl of soup or a cup of water to a person with reduced mobility [2], [3]. As far as robot manipulators are concerned, completely solving this planning problem is extremely complex and challenging [4]–[6]. The problem becomes even harder when environmental and object contacts have to be taken into account [7].

A two-step approach is usually used for time-optimal trajectory planning [8], [9]. In the first step, a collision-free path is planned, and in the second step, the dynamics of the system, as well as the actuator limits, are taken into account to convert the path to a minimum-time trajectory. In this paper, we focus on the second step for time-optimal object manipulation while considering contact constraints along with dynamic constraints, actuator limits, and any path constraints that might be present.

Contact with the environment during motion planning introduces additional complexities due to the underlying frictional dynamics [10]. However, contacts with the envi-

¹The authors are with the Department of Mechanical Engineering, Stony Brook University, USA. {dasharadhan.mahalingam, aditya.patankar, nilanjan.chakraborty}@stonybrook.edu.

²The authors are with Munich Institute of Robotics and Machine Intelligence (MIRMI), Technical University of Munich. Email:{riddhiman.laha, sami.haddadin}@tum.de

This work is partially supported by the US Department of Defense through ALSRP under Award No. HT94252410098.

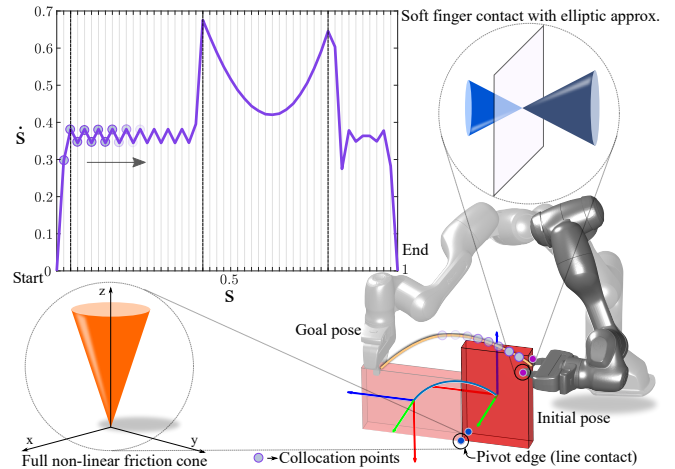


Fig. 1: Example pivoting task, which is a path constrained robot task involving manipulator object and object environment contacts. Our method takes into account all the kinematic and dynamic constraints of the system to compute the time-optimal trajectory. The (s, \dot{s}) phase plane, which connects the initial path position to the final position, illustrates our problem. The purple curve is the time scaling that we achieve while respecting position, velocity, and acceleration constraints. The blue dots denote the maximum feasible scaling possible at the collocation points. The vertical dashed lines represent specific instances when joints 6, 7, and 4 are about to hit the limits. Note that we consider the full non-linear friction cone constraints and the soft finger contact with elliptic approximations for modeling the two types of contacts.

ronment can be exploited to aid manipulation [11], [12]. For example, a *heavy* object can be manipulated by pivoting instead of lifting (Fig 1). Reaction forces generated at the environment-object contact can be used to partly balance the weight of the object. Therefore, our goal is to take contact constraints into account for trajectory optimization. To this end, we consider the complete nonlinear friction cone constraints instead of a polygonal approximation [13], [14]. This leads us to the problem we aim to focus: *How to bridge the gap between geometric motion planning and time-optimal trajectory planning such that the robot-object and environment-object contact constraints of the system are respected?*

Contributions: We present a novel second-order cone program (SOCP) formulation of the time-optimal trajectory planning problem that considers nonlinear friction cone constraints at the hand-object and object-environment contacts along with the constraints imposed by the robot dynamics and the robots’ actuator limits. Our model is general enough to handle multiple manipulators making contact with an object as well as multiple contacts between the object and the environment. Since SOCPs are convex optimization problems that can be solved optimally in polynomial time using interior point methods, we can solve the trajectory op-

timization problem efficiently. We present simulation results on three examples, including a non-prehensile manipulation task, which shows the generality and effectiveness of our approach.

II. RELATED WORK

The time-optimal trajectory planning problem for robotic manipulators has been a significant area of research in the literature. The core objective is to determine the optimal timing of motion along a predefined path, subject to various constraints, such as actuator torque limits and dynamic considerations. Traditionally, robot trajectory generation can be categorized into (a) offline and (b) online (real-time) strategies. Offline trajectory generation has a long history starting from the late '60s. Some of the seminal works in this regard are the solutions to minimum-time control problems [15], [16]. However, including collision-free trajectories in the introduced formulation is non-trivial.

The authors in [17] overcame this limitation by proposing a parameter optimization scheme that guaranteed minimum time travel using the maximum available torques while avoiding obstacles in the workspace. Similar directions were also explored in [18], where scaling was performed on existing path velocity profiles to maximize the output of the actuators. Similar algorithms were proposed in [5], [19], which exploited switching curves in the phase plane for end-effector motion along a path. The assumption for the latter one is that actuator torque bounds are a quadratic function of the joint velocity, with joint positions being parameterized by a polynomial path parameter. Recent approaches, in contrast, focus more on time-optimized path tracking, considering velocity, acceleration, and jerk limits [20]–[22]. Our approach, although similar to [23], differs considerably in the sense that we extend it to a more general scheme. Regardless, most of the works do not consider held objects and model explicitly the robot-object contact that we take care of during the motion execution. On the other hand, on-line paradigms fuse the path and time parameterization as a single step [24]. Typically, offline methods could be able to generate more optimal trajectories, although they are not suited for real-time applications owing to their computational load. In contrast, online methods are efficient, yet prone to discontinuities in the commanded trajectory [25].

The problem of computing optimal contact forces, at the manipulator-object contacts, was first formulated as a Second-Order Cone Program (SOCP) by [26]. A fast interior-point algorithm for solving this formulation was proposed in [27] with the point contact with friction model as the cone constraint. Building on this, the authors in [11] presented a SOCP for computing the optimal contact forces, while considering contact with the environment, for *quasi-statically* manipulating heavy objects by pivoting. The proposed formulation is general enough to incorporate manipulator joint limit constraints and has been integrated with a task space-based planning approach for pivoting [12]. In this work, we build upon the formulation proposed in [11] while considering the dynamics of the robot as well as the inertial properties of the object(s) being manipulated.

The authors in [28] formulate a *contact-implicit* trajectory optimization problem, as a nonlinear program, for the task of pushing an object along a plane surface to a desired goal

pose. The contact constraints have been represented as linear complementarity constraints which approximate the friction cones. Similar approaches [29]–[31] have been developed for tasks such as planar pushing in which an optimal sequence of control inputs has to be provided to make contact with the object and apply the necessary normal contact forces to push the object further away from the robot. In contrast, we are more interested in manipulation tasks which require the robot to maintain continuous contact, at more than one location, with the object throughout the motion.

III. SYSTEM MODELING AND ASSUMPTIONS

We consider v robot manipulators manipulating an object. Let there be u object-environment contacts. Coordinate frames $\{E_i\}$ (where $i = 1, 2, \dots, u$) and $\{M_j\}$ (where $j = 1, 2, \dots, v$) are fixed at each environment and manipulator contact respectively such that the z -axis of the coordinate frames are normal (inward) to the object surface and the other two axes, x and y are tangent to the surface. We assume that a collision-free geometric path is available as some waypoints for the robot manipulator which we want to convert to a time-optimal trajectory.

A. Manipulator Kinematics

Using the product of exponentials (POE) formula [13], the forward kinematics of an n DOF serial manipulator is

$$\mathbf{X} = e^{\hat{\xi}_1 q_1} e^{\hat{\xi}_2 q_2} \dots e^{\hat{\xi}_n q_n} \mathbf{X}_{ref}, \quad (1)$$

where $\xi_l \in se(3)$, $l = 1, 2, \dots, n$ are the twists contributed by each joint and $\mathbf{X}_{ref} \in SE(3)$ is the reference configuration of the manipulator at $\Theta = \mathbf{0}$. The body velocity of the end-effector of the manipulator, ${}^X \mathbf{V} \in \mathbb{R}^6$ is related to its joint rates through the body Jacobian $\mathbf{J}^b(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ as

$${}^X \mathbf{V} = \begin{bmatrix} \mathbf{v}^b \\ \boldsymbol{\omega}^b \end{bmatrix} = \mathbf{J}^b(\mathbf{q}) \dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_v^b(\mathbf{q}) \\ \mathbf{J}_\omega^b(\mathbf{q}) \end{bmatrix} \dot{\mathbf{q}}, \quad (2)$$

where \mathbf{v} and $\boldsymbol{\omega}$ denote the linear and angular velocities respectively, \mathbf{q} is the vector of joint displacements and $\dot{\mathbf{q}}$ is the vector of joint rates. The body acceleration of the end-effector of the manipulator can be determined from

$${}^X \dot{\mathbf{V}} = \begin{bmatrix} \dot{\mathbf{v}}^b \\ \dot{\boldsymbol{\omega}}^b \end{bmatrix} = \mathbf{J}^b(\mathbf{q}) \ddot{\mathbf{q}} + \dot{\mathbf{J}}^b(\mathbf{q}) \dot{\mathbf{q}}. \quad (3)$$

B. Manipulator Dynamics

Extending our analysis to the acceleration level, the equations of motion for the robot manipulator in contact with the environment can be written as,

$$\boldsymbol{\tau} - \mathbf{J}^T(\mathbf{q}) \mathbf{h}_e = \mathbf{M}(\mathbf{q}) \ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}), \quad (4)$$

where $\boldsymbol{\tau} \in \mathbb{R}^n$ is the vector of applied joint torques, $\mathbf{h}_e \in \mathbb{R}^6$ is the wrench exerted on the manipulator by the environment, $\mathbf{J}(\mathbf{q}) \in \mathbb{R}^{6 \times n}$ is the manipulator (geometric) Jacobian matrix, $\mathbf{M}(\mathbf{q}) \in \mathbb{R}^{n \times n}$ is the positive definite mass matrix, $\mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{n \times n}$ is the matrix accounting for Coriolis and centrifugal effects and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is the gravity term.

We are primarily interested in computing the joint torques required to produce the necessary wrenches at the manipulator-object contacts. Therefore, we need to establish a relationship between the joint torques and contact wrenches, \mathbf{F}_M exerted on the object. Let n_j be the number of DOFs of the j -th manipulator, $\mathbf{q}_j = [q_{j,1}, q_{j,2}, \dots, q_{j,n_j}]^T$ be the joint positions and $\boldsymbol{\tau}_j = [\tau_{j,1}, \tau_{j,2}, \dots, \tau_{j,n_j}]^T$ be the

joint torques of this manipulator. The equations of motion of the j -th manipulator can then be formulated as,

$$\boldsymbol{\tau}_j + \mathbf{J}_j^T(\mathbf{q}_j) \mathbf{f}_{M_j} = \mathbf{M}_j(\mathbf{q}_j) \ddot{\mathbf{q}}_j + \mathbf{C}_j(\mathbf{q}_j, \dot{\mathbf{q}}_j) \dot{\mathbf{q}}_j + \mathbf{g}_j(\mathbf{q}_j),$$

where $\mathbf{J}_j \in \mathbb{R}^{6 \times n_j}$ is the Jacobian matrix of the j -th manipulator expressed in the contact frame $\{M\}_j$ and $\mathbf{M}_j(\mathbf{q}_j)$, $\mathbf{C}_j(\mathbf{q}_j, \dot{\mathbf{q}}_j)$, $\mathbf{g}_j(\mathbf{q}_j)$ are the respective mass matrix, Coriolis and centrifugal effects, gravity term of the j -th manipulator. Here, we note that the wrench exerted on the manipulator is the reaction wrench $-\mathbf{f}_{M_j}$, and hence the $-\mathbf{J}_j^T(\mathbf{q}_j)(-\mathbf{f}_{M_j}) = +\mathbf{J}_j^T(\mathbf{q}_j) \mathbf{f}_{M_j}$ term on the left side of the equation. By defining $\mathbf{q} \in \mathbb{R}^n$ and $\boldsymbol{\tau} \in \mathbb{R}^n$ ($n = \sum_{j=1}^v n_j$) as $\mathbf{q} = [\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_v]^T$ and $\boldsymbol{\tau} = [\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \dots, \boldsymbol{\tau}_v]^T$, the relationship between the joint torques of all the manipulators ($\boldsymbol{\tau}$) and all the manipulator-object contact wrenches (\mathbf{F}_M) can be represented in compact form as

$$\boldsymbol{\tau} + \mathcal{J}_M(\mathbf{q}) \mathbf{F}_M = \mathcal{M}_M(\mathbf{q}) \ddot{\mathbf{q}} + \mathcal{C}(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathcal{G}(\mathbf{q}). \quad (5)$$

where $\mathcal{J}_M = \text{diag}(\mathbf{J}_1^T, \mathbf{J}_2^T, \dots, \mathbf{J}_v^T) \in \mathbb{R}^{n \times 6v}$ is the overall Jacobian matrix of all the manipulators, $\mathcal{M}_M = \text{diag}(\mathbf{M}_1, \mathbf{M}_2, \dots, \mathbf{M}_v) \in \mathbb{R}^{n \times n}$ is the overall mass matrix of all the manipulators, $\mathcal{C}_M = \text{diag}(\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_v) \in \mathbb{R}^{n \times n}$ is the overall Coriolis and centrifugal effect of all the manipulators and $\mathcal{G}_M = [\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_v]^T \in \mathbb{R}^n$ is the overall gravity term of all the manipulators.

C. Object Dynamics

Let ${}^O\mathbf{R}_{E_i} \in SO(3)$ and ${}^O\mathbf{R}_{M_j} \in SO(3)$ be 3×3 orthogonal matrices representing the orientation of the *environment-object contact* coordinate frame and *manipulator-object contact* coordinate frame respectively expressed in the object coordinate frame $\{O\}$. Each contact wrench, \mathbf{f}_{E_i} and \mathbf{f}_{M_j} , can be expressed in the object coordinate frame $\{O\}$ as

$${}^O\mathbf{f}_{E_i} = \mathbf{G}_{E_i} \mathbf{f}_{E_i} = \begin{bmatrix} {}^O\mathbf{R}_{E_i} & \mathbf{0} \\ \mathbf{S}({}^O\mathbf{p}_{E_i}) {}^O\mathbf{R}_{E_i} & {}^O\mathbf{R}_{E_i} \end{bmatrix} \mathbf{f}_{E_i}, \quad (6)$$

$${}^O\mathbf{f}_{M_j} = \mathbf{G}_{M_j} \mathbf{f}_{M_j} = \begin{bmatrix} {}^O\mathbf{R}_{M_j} & \mathbf{0} \\ \mathbf{S}({}^O\mathbf{p}_{M_j}) {}^O\mathbf{R}_{M_j} & {}^O\mathbf{R}_{M_j} \end{bmatrix} \mathbf{f}_{M_j} \quad (7)$$

Here, $\mathbf{S}(\cdot) \in skew(3)$ is the skew-symmetric operator [32], such that, for any $\mathbf{a} \in \mathbb{R}^3, \mathbf{b} \in \mathbb{R}^3$, $\mathbf{S}(\mathbf{a})\mathbf{b} = \mathbf{a} \times \mathbf{b}$. Also, ${}^O\mathbf{p}_{E_i}$ and ${}^O\mathbf{p}_{M_j}$ are the positions of $\{E\}_i$ and $\{M\}_j$ expressed in $\{O\}$ respectively. Therefore, the object motion satisfies the equation,

$$\sum_{i=1}^u \mathbf{G}_{E_i} \mathbf{f}_{E_i} + \sum_{j=1}^v \mathbf{G}_{M_j} \mathbf{f}_{M_j} + {}^O\mathbf{f}_{ext} = {}^O\mathbf{f}_{net}. \quad (8)$$

where ${}^O\mathbf{f}_{ext} \in \mathbb{R}^6$ is the total external (force and moment) wrench acting on the object (including the object weight) expressed in O and ${}^O\mathbf{f}_{net}$ is the net wrench (resultant of all wrenches) acting on the object expressed in O .

By introducing $\mathbf{F}_E \in \mathbb{R}^{6u}$ and $\mathbf{F}_M \in \mathbb{R}^{6v}$ as $\mathbf{F}_E = [\mathbf{f}_{E_1}, \mathbf{f}_{E_2}, \dots, \mathbf{f}_{E_u}]^T$ and $\mathbf{F}_M = [\mathbf{f}_{M_1}, \mathbf{f}_{M_2}, \dots, \mathbf{f}_{M_v}]^T$ equation (8) can be represented in a more compactly as,

$$\mathbf{G}_E \mathbf{F}_E + \mathbf{G}_M \mathbf{F}_M + {}^O\mathbf{f}_{ext} = {}^O\mathbf{f}_{net} \quad (9)$$

where $\mathbf{G}_E = [\mathbf{G}_{E_1}, \mathbf{G}_{E_2}, \dots, \mathbf{G}_{E_u}] \in \mathbb{R}^{6 \times 6u}$ is the environment contact matrix and $\mathbf{G}_M = [\mathbf{G}_{M_1}, \mathbf{G}_{M_2}, \dots, \mathbf{G}_{M_v}] \in \mathbb{R}^{6 \times 6v}$ is the manipulator contact matrix, which is also known as the *grasp map* or *grasp matrix* [32].

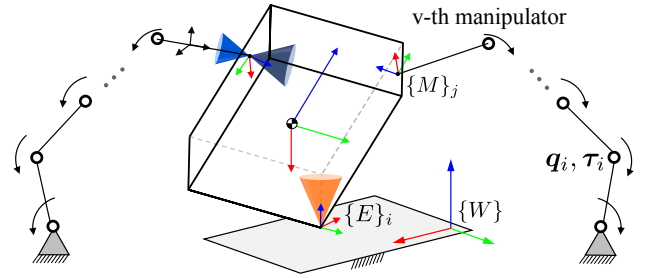


Fig. 2: A rigid body being manipulated with v manipulators and in contact with the environment at u points. The contacts between object and the environment are modeled as PCWF and contacts between manipulators and the objects are assumed to be SFCE. $\{E\}$, $\{M\}$, and $\{W\}$ represent the environment, manipulator, and world reference frames.

D. Contact Constraints

We use the point contact with friction (PCWF) for the object-environment contacts, and the soft finger contact with elliptic approximation (SFCE) for the contacts between the manipulators and the object [11]. For PCWF, the contact wrench satisfies the friction cone constraint as

$$\frac{1}{\mu_{E_i}} \sqrt{\left(\frac{f_{E_i,x}}{e_{E_i,x}}\right)^2 + \left(\frac{f_{E_i,y}}{e_{E_i,y}}\right)^2} \leq f_{E_i,z}, \quad (10)$$

where the parameters $\mu_{E_i}, e_{E_i,x}$ and $e_{E_i,y}$ are positive constants defining the friction coefficients at the i -th environment contact point and $i = 1, 2, \dots, u$. Similarly, in SFCE, the contact wrench satisfies the elliptic constraint as

$$\frac{1}{\mu_{M_j}} \sqrt{\left(\frac{f_{M_j,x}}{e_{M_j,x}}\right)^2 + \left(\frac{f_{M_j,y}}{e_{M_j,y}}\right)^2 + \left(\frac{\tau_{M_j,z}}{e_{M_j,z}}\right)^2} \leq f_{M_j,z}, \quad (11)$$

where the parameters $\mu_{M_j}, e_{M_j,x}, e_{M_j,y}, e_{M_j,z}$ are positive constants defining the friction coefficients at the j -th manipulator contact and $j = 1, 2, \dots, v$. The constraints (10) and (11) are *Second-Order Cone* (SOC) constraints [33]. For any $\mathbf{r} \in \mathbb{R}^6$, we define the friction cones \mathcal{K}_{E_i} and \mathcal{K}_{M_j} as

$$\mathcal{K}_{E_i} = \left\{ \mathbf{r} \mid \frac{1}{\mu_{E_i}} \sqrt{\frac{r_1^2}{e_{E_i,x}^2} + \frac{r_2^2}{e_{E_i,y}^2}} \leq r_3, r_{4,5,6} = 0 \right\},$$

$$\mathcal{K}_{M_j} = \left\{ \mathbf{r} \mid \frac{1}{\mu_{M_j}} \sqrt{\frac{r_1^2}{e_{M_j,x}^2} + \frac{r_2^2}{e_{M_j,y}^2} + \frac{r_6^2}{e_{M_j,z}^2}} \leq r_3, r_{4,5} = 0 \right\},$$

where $r_k, k = 1, 2, \dots, 6$ is the k -th component of \mathbf{r} . To ensure the manipulated object is not damaged by the internal forces, we consider an upper bound \bar{F}_z on the normal contact force. Thus, the normal contact force $\mathbf{F}_z = [f_{M_{1,z}}, f_{M_{2,z}}, \dots, f_{M_{v,z}}]^T \leq \bar{F}_z^1$.

IV. MATHEMATICAL PROBLEM FORMULATION

A. Time Parameterization of Motion

Let us assume that the object frame $\{O\}$ goes through a trajectory which can be expressed using a scalar path coordinate s , as $\{O\}(s)$. Assuming that the object starts moving at time, $t = 0$ and finishes moving through the trajectory at time, $t = T$, the time dependence of the object trajectory can be obtained through the time scaling $s(t)$ which assigns a value to s for any instant of time $t \in [0, T]$,

¹Note that a set of contact forces that result in no net force on object is termed as *internal forces*.

such that $s : [0, T] \mapsto [0, 1]$ with $s(0) = 0$ and $s(T) = 1$. For notational convenience, let us omit the time dependence of the path coordinate s and its derivatives.

From the motion of $\{O\}$, we can then determine the joint-space path that all the manipulators take as $\mathbf{q}(s)$, which is a function of the scalar path coordinate s . The joint velocities and joint accelerations can then be obtained as $\dot{\mathbf{q}}(s) = \frac{d\mathbf{q}(s)}{ds} \frac{ds}{dt} = \mathbf{q}'(s)\dot{s}$ and $\ddot{\mathbf{q}}(s) = \mathbf{q}''(s)\dot{s}^2 + \mathbf{q}'(s)\ddot{s}$, where, $\dot{s} = ds/dt$, $\ddot{s} = d^2s/dt^2$, $\mathbf{q}'(s) = d\mathbf{q}(s)/ds$, and $\mathbf{q}''(s) = d^2\mathbf{q}(s)/ds^2$.

The equations of motion of the object can also be described using the scalar path coordinate s as

$$\mathbf{G}_E(s)\mathbf{F}_E(s) + \mathbf{G}_M\mathbf{F}_M(s) + {}^O\mathbf{f}_{ext}(s) = {}^O\mathbf{f}_{net}(s). \quad (12)$$

Please note that, while the environment contact matrix $\mathbf{G}_E(s)$ may change depending upon the motion of the object. However, the manipulator contact matrix \mathbf{G}_M remains constant as long as contact constraints at the manipulator-object contacts are satisfied. Therefore by incorporating the friction cone constraints at the manipulator-object contact, the object can be considered as an extension of the manipulator's end-effector. This assumption lets us express the object's position, velocity and acceleration expressed in its reference frame O by relating it to the j -th manipulator's end-effector velocity and acceleration as,

$${}^O\mathbf{X}(s) = \mathbf{X}_j^{M_j}\mathbf{T}_O, \quad (13)$$

$${}^O\mathbf{V}(s) = \mathbf{J}_j^b(\mathbf{q}_j(s))\dot{\mathbf{q}}_j(s) = \mathbf{J}_j^b(\mathbf{q}_j(s))\mathbf{q}'(s)\dot{s}, \quad (14)$$

$${}^O\dot{\mathbf{V}}(s) = \frac{d(\mathbf{J}_j^b)}{dt}\dot{\mathbf{q}}_j(s) + \mathbf{J}_j^b(\mathbf{q}_j(s))\ddot{\mathbf{q}}_j(s), \quad (15)$$

$${}^O\dot{\mathbf{V}}(s) = (\mathbf{J}_j^{b'}(s)\mathbf{q}'_j(s) + \mathbf{J}_j^b(\mathbf{q}_j(s))\mathbf{q}''_j(s))\dot{s}^2 + \mathbf{J}_j^b(\mathbf{q}_j(s))\mathbf{q}'_j(s)\ddot{s}, \quad (16)$$

where, ${}^{M_j}\mathbf{T}_O$ is the relative rigid body transformation expressing the pose of the object frame O in the manipulator's end-effector frame and,

$$\mathbf{J}_j^{b'}(s) = \sum_{i=1}^{n_j} \frac{d(\mathbf{J}_j^b(\mathbf{q}_j(s)))}{dq_{j,i}} \frac{dq_{j,i}}{ds}. \quad (17)$$

We note here that the relative rigid body transformation between the j -th manipulator's end-effector reference frame $\{X_j\}$ and the object reference frame $\{O\}$ needs to be taken into account when computing the body Jacobian \mathbf{J}_j^b for this to be true. Hereafter, we will use the 1-st manipulator to determine ${}^O\mathbf{V}$ and ${}^O\dot{\mathbf{V}}$. However any one of the v manipulators can be used for this purpose.

$${}^O\mathbf{V}(s) = \mathcal{J}_O(s)\dot{s}, \quad {}^O\dot{\mathbf{V}}(s) = \mathcal{J}'_O(s)\dot{s}^2 + \mathcal{J}_O(s)\ddot{s}, \quad (18)$$

$$\mathcal{J}_O(s) = \mathbf{J}_1^b(\mathbf{q}_1(s))\mathbf{q}'_1(s), \quad (19)$$

$$\mathcal{J}'_O(s) = \mathbf{J}_1^{b'}(s)\mathbf{q}'_1(s) + \mathbf{J}_1^b(\mathbf{q}_1(s))\mathbf{q}''_1(s), \quad (20)$$

The net wrench acting on the object whose mass is m kg and mass moment of inertia is ${}^O\mathcal{I}$ can then be expressed as

$${}^O\mathbf{f}_{net}(s) = \mathcal{M}_O{}^O\dot{\mathbf{V}}(s) + \mathcal{C}_O(s)\dot{s}^2, \quad (21)$$

$${}^O\mathbf{f}_{net}(s) = \mathcal{M}_O\mathcal{J}_O(s)\ddot{s} + (\mathcal{M}_O\mathcal{J}'_O(s) + \mathcal{C}_O(s))\dot{s}^2, \quad (22)$$

Here, $\mathcal{M}_O = \begin{bmatrix} \text{diag}(m) & \mathbf{0} \\ \mathbf{0} & {}^O\mathcal{I} \end{bmatrix}$ is the object mass matrix and is a constant in the object reference frame and \mathcal{C}_O is the

Coriolis and centrifugal term of the object which is given by

$$\mathcal{C}_O(s)\dot{s}^2 = \begin{bmatrix} \boldsymbol{\omega}^b(s) \times m\mathbf{v}^b(s) \\ \boldsymbol{\omega}^b(s) \times {}^O\mathcal{I}\boldsymbol{\omega}^b(s) \end{bmatrix}, \quad (23)$$

$$\mathcal{C}_O(s)\dot{s}^2 = \begin{bmatrix} \mathbf{S}(\mathcal{J}_{O_\omega}(s)) & \mathbf{0} \\ \mathbf{0} & \mathbf{S}(\mathcal{J}_{O_\omega}(s)) \end{bmatrix} \mathcal{M}_O\mathcal{J}_O(s)\dot{s}^2, \quad (24)$$

The friction cone constraints can also be described using the scalar path coordinate s as

$$\mathbf{f}_{E_i}(s) \in \mathcal{K}_{E_i}, i = 1, 2, \dots, u \quad (25)$$

$$\mathbf{f}_{M_j}(s) \in \mathcal{K}_{M_j}, j = 1, 2, \dots, v \quad (26)$$

The upper bound on the normal contact forces can be described using the scalar path coordinates s as

$$\mathbf{F}_z(s) \leq \bar{\mathbf{F}}_z(s) \quad (27)$$

Substituting the expressions for $\dot{\mathbf{q}}(s)$ and $\ddot{\mathbf{q}}(s)$ in (5), we get the following

$$\boldsymbol{\tau}(s) + \mathcal{J}_M(s)^T\mathbf{F}_M(s) = \mathcal{M}_M(s)\ddot{s} + \mathcal{C}_M(s)\dot{s}^2 + \mathcal{G}_M(s), \quad (28)$$

where

$$\mathcal{M}_M(s) = \mathbf{M}(\mathbf{q}(s))\mathbf{q}'(s), \quad \mathcal{G}_M(s) = \mathbf{G}(\mathbf{q}(s)), \quad (29)$$

$$\mathcal{C}_M(s) = \mathbf{M}(\mathbf{q}(s))\mathbf{q}''(s) + \mathbf{C}(\mathbf{q}(s), \mathbf{q}'(s))\mathbf{q}'(s). \quad (30)$$

To also ensure that the joint actuators of the manipulators are within the actuator limits and not saturated, the following joint torque limit constraints need to be considered

$$\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}(s) \leq \bar{\boldsymbol{\tau}} \quad (31)$$

where $\underline{\boldsymbol{\tau}}$ and $\bar{\boldsymbol{\tau}}$ are the lower and upper joint torque limits of the manipulators respectively.

If, the equations (12), (25), (26), (27), (28), (31) are satisfied simultaneously throughout the motion, then the object can be manipulated while satisfying the contact constraints.

B. Time Optimal Control

The extension of the problem of time-optimal control of a manipulator along a path, as in [23], applied to v manipulators with the added contacts can be expressed as,

$$\underset{T, s, \boldsymbol{\tau}, \mathbf{F}_E, \mathbf{F}_M}{\text{minimize}} T \quad (32)$$

subject to (for $t \in [0, T]$),

$$\boldsymbol{\tau}(s) + \mathcal{J}_M(s)\mathbf{F}_M(s) = \mathcal{M}_M(s)\ddot{s} + \mathcal{C}_M(s)\dot{s}^2 + \mathcal{G}_M(s)$$

$$\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}(s) \leq \bar{\boldsymbol{\tau}}$$

$$\mathbf{G}_E(s)\mathbf{F}_E(s) + \mathbf{G}_M\mathbf{F}_M(s) + {}^O\mathbf{f}_{ext}(s) = {}^O\mathbf{f}_{net}(s)$$

$${}^O\mathbf{f}_{net}(s) = \mathcal{M}_O\mathcal{J}_O(s)\ddot{s} + (\mathcal{M}_O\mathcal{J}'_O(s) + \mathcal{C}_O(s))\dot{s}^2$$

$$\mathbf{f}_{E_i}(s) \in \mathcal{K}_{E_i}, i = 1, 2, \dots, u$$

$$\mathbf{f}_{M_j}(s) \in \mathcal{K}_{M_j}, j = 1, 2, \dots, v$$

$$\mathbf{F}_z(s) \leq \bar{\mathbf{F}}_z$$

$$s(0) = 0, s(T) = 1, \dot{s}(0) = \dot{s}_0, \dot{s}(T) = \dot{s}_T, \dot{s}(t) \geq 0$$

C. Reformulation as a Convex Optimization Problem

We adopt the strategy of introduction of new optimization variables $a(s), b(s)$, as shown in [23], to obtain a reformulated objective function as

$$T = \int_0^T dt = \int_0^1 \frac{1}{\dot{s}} ds \quad (33)$$

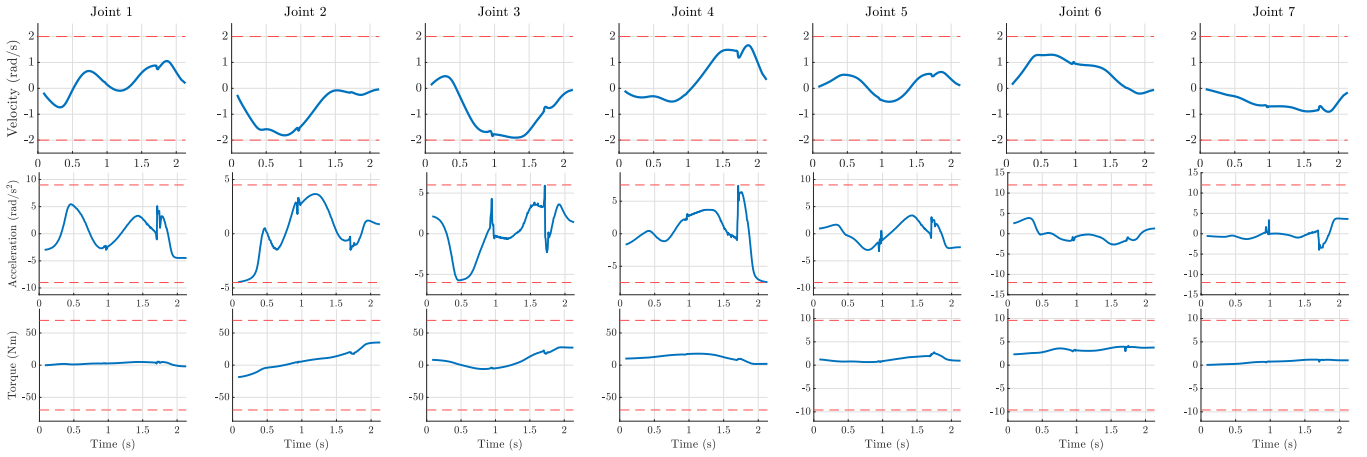


Fig. 3: Plots of joint velocities, accelerations and torques for the task of picking up an object. The joint limits are shown using dashed red lines. Joints 3 and 4 are the ones getting closest to their velocity and acceleration limits.

The new optimization variables are $a(s) = \ddot{s}$, $b(s) = \dot{s}^2$, and by adding the constraint $b'(s) = 2a(s)$ results in,

$$\dot{b}(s) = 2\dot{s}\ddot{s} = 2\dot{s}a(s) \quad (34)$$

Thus, the optimization problem now becomes

$$\underset{a,b,\tau,\mathbf{F}_E,\mathbf{F}_M}{\text{minimize}} \int_0^1 \frac{1}{\sqrt{b(s)}} ds \quad (35)$$

subject to (for $s \in [0, 1]$),

$$\boldsymbol{\tau}(s) + \mathcal{J}_M(s)\mathbf{F}_M(s) = \mathcal{M}_M(s)a(s) + \mathcal{C}_M(s)b(s) + \mathcal{G}_M(s)$$

$$\underline{\boldsymbol{\tau}} \leq \boldsymbol{\tau}(s) \leq \overline{\boldsymbol{\tau}}$$

$$\mathbf{G}_E(s)\mathbf{F}_E(s) + \mathbf{G}_M\mathbf{F}_M(s) + {}^O\mathbf{f}_{ext}(s) = {}^O\mathbf{f}_{net}(s)$$

$${}^O\mathbf{f}_{net}(s) = \mathcal{M}_O\mathcal{J}_O(s)a(s) + (\mathcal{M}_O\mathcal{J}'_O(s) + \mathcal{C}_O(s))b(s)$$

$$\mathbf{f}_{E_i}(s) \in \mathcal{K}_{E_i}, i = 1, 2, \dots, u$$

$$\mathbf{f}_{M_j}(s) \in \mathcal{K}_{M_j}, j = 1, 2, \dots, v$$

$$\mathbf{F}_z(s) \leq \overline{\mathbf{F}}_z$$

$$b(0) = (\dot{s}(0))^2, b(T) = (\dot{s}(T))^2, b(s) \geq 0, b'(s) = 2a(s)$$

Here, the objective function is convex and all the constraints except for the SOC constraints \mathcal{K}_{E_i} and \mathcal{K}_{M_j} are linear functions of the optimization variables. Thus, the above optimization problem is a convex optimization problem.

D. Manipulator Motion Constraints

Apart from the actuator force/torque limits there can be other constraints such as velocity and acceleration limits in the joint space as well as in the task space which are convex constraints and can be included in the optimization. However, here we focus only on the joint space constraints.

Joint velocity constraints: Linear inequality constraints

$$0 \leq \dot{q}_{j,l}^2(s) \leq \overline{q}_{j,l}^2 \quad (36)$$

$$0 \leq q'_{j,l}(s)\dot{s} \leq \overline{q}_{j,l}^2 \quad (37)$$

$$0 \leq q'_{j,l}(s)b(s) \leq \overline{q}_{j,l}^2 \quad (38)$$

for all $j = 1, \dots, v$ and $l = 1, \dots, n_j$

Joint acceleration constraints: Linear inequality constraints

$$\underline{\ddot{q}} \leq \ddot{q}(s) \leq \overline{\ddot{q}} \quad (39)$$

$$\underline{\ddot{q}} \leq \mathbf{q}''(s)\dot{s}^2 + \mathbf{q}'(s)\ddot{s} \leq \overline{\ddot{q}} \quad (40)$$

$$\underline{\ddot{q}} \leq \mathbf{q}''(s)b(s) + \mathbf{q}'(s)a(s) \leq \overline{\ddot{q}} \quad (41)$$

V. IMPLEMENTATION AND SOLUTION APPROACH

In this section, we detail the solution strategy for the optimization problem formulated in the previous section.

A. Direct Transcription

We follow the collocation approach proposed in [23] where $a(s)$ is assumed to be the piece-wise constant control input. This implies that $b(s)$ is piecewise linear and $\tau(s)$ is piecewise nonlinear. Thus, based on the values at the grid points, $b(s)$ becomes

$$b(s) = b^k + \left(\frac{b^{k+1} - b^k}{s^{k+1} - s^k} \right) (s - s^k) \text{ for } s \in [s^k, s^{k+1}]$$

Let us define a^{k_m} , τ^{k_m} , $\mathbf{F}_E^{k_m}$, $\mathbf{F}_M^{k_m}$ be the values of the functions $a(s)$, $\tau(s)$, $\mathbf{F}_E^k(s)$, $\mathbf{F}_M^k(s)$ at $s^{k_m} = \frac{s^{k+1} + s^k}{2}$, which is the midpoint of the grid points s^k and s^{k+1} .

The objective function can now be expressed as

$$\int_0^1 \frac{1}{\sqrt{b(s)}} ds = \sum_{k=0}^{K-1} \int_{s^k}^{s^{k+1}} \frac{1}{\sqrt{b(s)}} ds = \sum_{k=0}^{K-1} \frac{2(\Delta s^k)}{\sqrt{b^{k+1}} + \sqrt{b^k}} \quad (42)$$

where $\Delta s^k = s^{k+1} - s^k$.

However, this problem can be reformulated as a Second Order Cone Program (SOCP) and it is advantageous to do so as there are very effective methods for solving them [27].

B. Second Order Cone Program

The above direct transcription problem can be reformulated as a SOCP by introducing variables c^k and d^k with the addition of the following inequality constraints [23]

$$c^k \leq \sqrt{b^k}, k = 0, \dots, K \quad (43)$$

$$\frac{1}{c^{k+1} + c^k} \leq d^k, k = 0, \dots, K-1 \quad (44)$$

These constraints can also be expressed as SOC constraints

$$\left\| \begin{matrix} 2c^k \\ b^k - 1 \end{matrix} \right\| \leq b^k + 1 \quad (45)$$

$$\left\| \begin{matrix} 2 \\ c^{k+1} + c^k - d^k \end{matrix} \right\| \leq c^{k+1} + c^k + d^k \quad (46)$$

This change in variable along with the introduction of the SOC constraints results in the following SOCP

$$\underset{a^k, b^k, c^k, d^k, \tau^{k_m}, \mathbf{F}_E^{k_m}, \mathbf{F}_M^{k_m}}{\text{minimize}} \quad \sum_{k=0}^{K-1} 2(\Delta s^k) d^k \quad (47)$$

subject to,

$$\begin{aligned} \tau^{k_m} + \mathcal{J}_M^{k_m} \mathbf{F}_M^{k_m} &= \mathcal{M}_M^{k_m} a^{k_m} + \mathcal{C}^{k_m} b^{k_m} + \mathcal{G}^{k_m} \\ \underline{\tau} &\leq \tau^{k_m} \leq \bar{\tau} \\ 0 \leq (q_{j,l}^{k_m})^2 b^{k_m} &\leq \bar{q}_{j,l}^2 \quad \forall j = 1, \dots, v \text{ and } l = 1, \dots, n_j \\ \underline{q} &\leq \mathbf{q}^{k_m} b^{k_m} + \mathbf{q}'^{k_m} a^{k_m} \leq \bar{q} \\ \mathbf{G}_E^{k_m} \mathbf{F}_E^{k_m} + \mathbf{G}_M \mathbf{F}_M^{k_m} + \mathbf{O} \mathbf{f}_{ext}^{k_m} &= \mathbf{O} \mathbf{f}_{net}^{k_m} \\ \mathbf{O} \mathbf{f}_{net}^{k_m}(s) &= \mathcal{M}_O \mathcal{J}_O^{k_m} a^{k_m} + (\mathcal{M}_O \mathcal{J}_O'^{k_m} + \mathcal{C}_O^{k_m}) b^{k_m} \\ \mathbf{f}_{E_i}^{k_m} &\in \mathcal{K}_{E_i}, i = 1, 2, \dots, u, \mathbf{f}_{M_j}^{k_m} \in \mathcal{K}_{M_j}, j = 1, 2, \dots, v \\ \mathbf{F}_z^{k_m} \leq \bar{\mathbf{F}}_z, \left\| \begin{matrix} 2 \\ c^{k+1} + c^k - d^k \end{matrix} \right\| &\leq c^{k+1} + c^k + d^k \end{aligned}$$

for $k = 0 \dots K - 1$,

$$\left\| \begin{matrix} 2c^k \\ b^k - 1 \end{matrix} \right\| \leq b^k + 1, b^k \geq 0, k = 0 \dots, K \\ b^0 = (\dot{s}^0)^2, b^K = (\dot{s}^K)^2$$

If $n = \sum_{j=1}^v n_j$ is the sum of the DOFs of all the v manipulators, then for a choice of K grid points, the total number of optimization variables is $K(4 + 3u + 4v + n) - 2$.

VI. SIMULATION STUDIES

We now present simulation experiments to evaluate our proposed formulation. We use YALMIP [34] which is a MATLAB toolbox for modeling and optimization with SeDuMi [35] as the solver. All experiments were carried out on a laptop running Ubuntu 20.04, equipped with an Intel Core i9 – 9880H CPU having 16 cores. The 7 DOF Frank Emika Panda research robot with a two finger parallel jaw gripper (see Figures 1 and 6) is used as the experimental test-bed. For all experiments, we set the torque bounds of the manipulator to be 80% of the manipulator’s rated torque limits² and acceleration bounds to be 60% of the manipulator’s rated acceleration limits (see datasheet [36] for robot hardware limits). We use a Screw Linear Interpolation (ScLERP) based motion planner with Jacobian pseudoinverse [37] to determine the joint space motion plan that the robot needs to execute. A key fact to note is that the optimization problem does not modify the joint space path. It only solves for the time-parameterization of the path parameter $s(t)$.

We use 3 exemplar tasks for validation of our approach: (a) pivoting an object while maintaining continuous contact with environment, (b) picking up an object from a flat surface, and (c) a non-prehensile manipulation task where the robot manipulates an object on a tray, which is called the ‘General Waiter Motion Problem’. We demonstrate that if there exists a minimum execution time solution that can satisfy the dynamic constraints, then by solving this optimization problem, we can determine the time parameterization of the path parameter $s(t)$, joint velocities $\dot{\mathbf{q}}(s(t))$ joint accelerations

$\ddot{\mathbf{q}}(s(t))$, torques $\tau(s(t))$, manipulator-object forces $\mathbf{F}_M(s(t))$ and object-environment contact forces $\mathbf{F}_E(s(t))$, such that the trajectory execution time is a minimum. This is due to the fact that the optimization problem is convex, and hence any local optimum is also the global optimum.

A. Use-Case #1: Time-Optimal Object Pivoting

The objective of this task is to use a manipulator to successfully pivot an cuboidal object while maintaining continuous contact with the environment (Fig. 1). The mass of the cuboid was set as 2 kg, the values of the friction parameters were chosen as $\mu_M = 0.4$, $e_{E_*} = e_{M_x} = e_{M_y} = 1$, $e_{M_z} = 0.25$, and the number of grid points was chosen as $K = 250$. \mathbf{F}_z was set as 70 N for all manipulator-object contacts and the joint space velocity bounds were set to ± 1 rad/s for all joints. We consider two manipulator-object contacts and two environment-object contacts at the vertices of the pivoting edge (line contact). Although the formulation given in (47) considers only one contact per manipulator, it can also be generalized to w manipulator-object contacts for the j -th manipulator by introducing a friction cone constraint (11) for each contact. We carried out multiple trials with values of μ_E 0.2, 0.3, 0.4 and 0.5. In all trials, the manipulator started at the same configuration and the object paths are the same. The joint velocities obtained for the pivoting task with $\mu_E = 0.4$ are shown in Fig 4. We can see that the joints 6, 7 and 4 reach the velocity limits during execution. Due to the velocity limits being reached in all the trials and identical paths for the object and manipulator, we observed the same optimal trajectory execution time of 2.5972 seconds across all the trials. However, if we increase the mass of the object beyond a certain value (depending on the choice of μ_E), we get infeasibility, meaning that the manipulator cannot pivot the object within the given actuator/contact force limits. Note that it takes approximately 2.5 seconds to obtain the solution.

B. Use-Case #2: Minimum Time Object Pick-up

We evaluate our proposed approach for the task of picking up a object (that is of cuboid shape) from a flat surface as shown in Fig 6. Note that for this task we do not consider any contact of the object with the environment. We only consider the two manipulator-object contacts (see Figure 6). The number of grid points was chosen as $K = 250$, the mass of the cuboid was set as 1 kg and the values of the friction parameters were chosen to be $\mu_M = 0.6$, $e_{M_x} = e_{M_y} = 1$, $e_{M_z} = 0.25$. \mathbf{F}_z was set as 70 N for all manipulator-object contacts and the joint space velocity bounds were set to ± 2 rad/s for all joints. We carried out multiple trials with the values for object mass m (kg) from $\{0.5, 0.75, 1.0, 1.25, 1.5, 1.75\}$. We observed an increase in the optimal trajectory execution time of 1.99 seconds for $m = 0.5$ kg upto 3.2 seconds for $m = 1.25$ kg. Since we constrain the maximum magnitude of normal force that can be applied at the manipulator-object contacts, the problem becomes infeasible for $m \geq 1.5$ kg. The results shown in Fig 3 are for $m = 1$ kg. After modeling, YALMIP took approximately 1.5 seconds for generating the output.

C. Use-Case #3: General Waiter Motion Problem

We evaluate our approach for the general waiter motion problem which is an instance of non-prehensile manipula-

² \mathbf{F}_z was set to be less than or equal to the maximum continuous grasping force of the Franka Hand gripper.

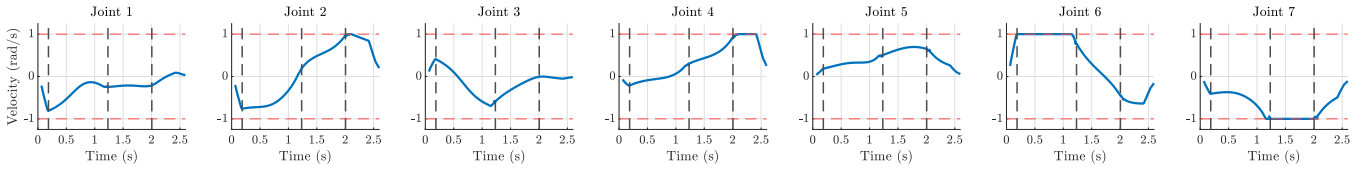


Fig. 4: Plots of joint velocities for the pivoting task. The vertical dashed lines denote specific instances when the joints are at their limits and they correspond to the vertical dashed lines shown in the (s, \dot{s}) phase plane in Fig 1.

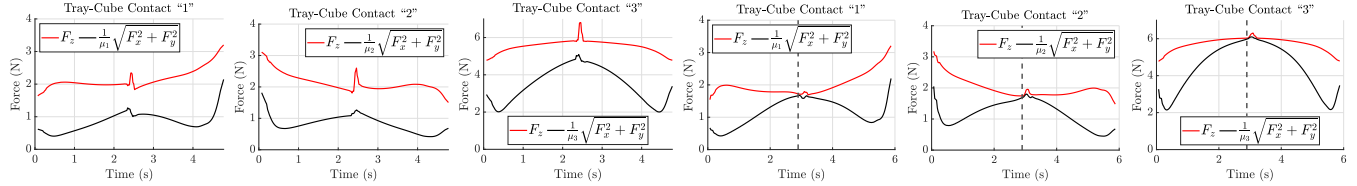


Fig. 5: Tray-Cube Friction Cone Constraint results obtained for the non-prehensile task. First three images on the left show the constraints when the tray angle was set to 10° . The next three images show the constraints when the tray angle was set to 15° . The vertical dashed line in the last three images depict the constraints at their limits.

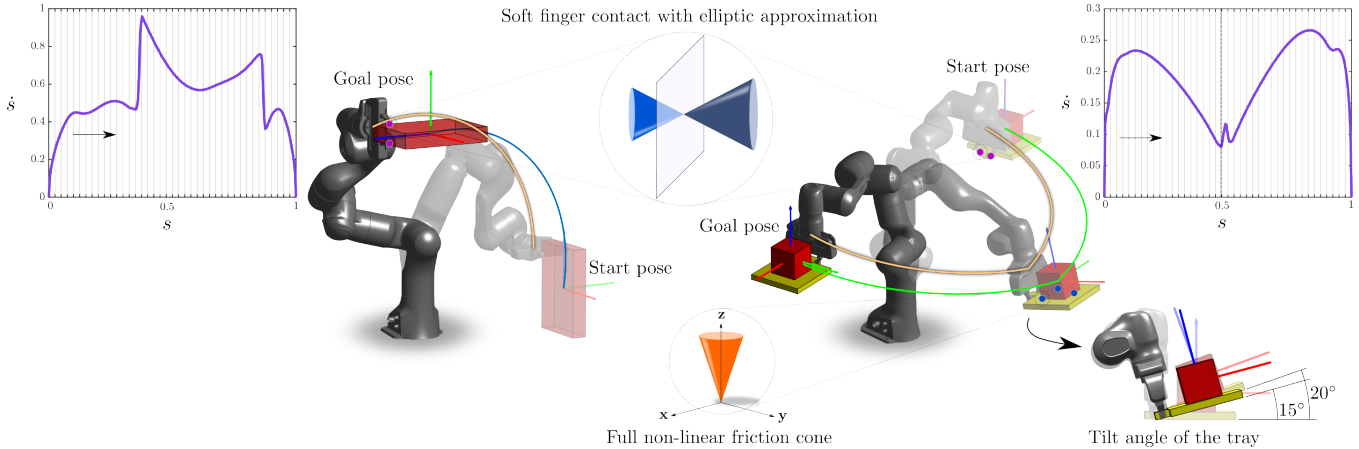


Fig. 6: **Left:** Picking up an object along a constrained path in a minimum time. **Right:** Non-prehensile manipulation task: Waiter Motion Problem. The phase plane at the top left and right corners depict the limits of the system in the two use-cases as the scaling evolves.

tion. The goal of the task is to move the tray with the cube on top from the starting pose to the end pose on the desired trajectory as fast as possible (Fig 6).

Although this problem has been widely studied in the context of trajectory optimization [38]–[41], none of the approaches takes into account contact constraints. Of the few that do [14], the usual norm is to approximate the friction cones with polygons for fast computation. Our approach, on the other hand, is general enough to take into account contact constraints between the cube and the tray as well (Eq. 47). Since we use a parallel jaw gripper to grasp the tray, we introduce two manipulator-tray contacts (SFCE model). The surface contact between the tray and the cube is modeled using three point contacts (see Fig 6). Consequently, there are three tray-cube contacts (PCWF model).

Since we have two objects, we have two associated equations of motion. The equations of motion for the cube are

$${}^{O_C} \mathbf{f}_{net}(s) = M_{O_C} \mathcal{J}_{O_C}(s) \ddot{s} + (M_{O_C} \mathcal{J}'_{O_C}(s) + C_{O_C}(s)) \dot{s}^2$$

$$\mathbf{G}_{CT}(s) \mathbf{F}_{CT}(s) + {}^{O_C} \mathbf{f}_{C,ext}(s) = {}^{O_C} \mathbf{f}_{C,net}(s)$$

and the equations of motion associated with the tray are

$${}^{O_T} \mathbf{f}_{net}(s) = M_{O_T} \mathcal{J}_{O_T}(s) \ddot{s} + (M_{O_T} \mathcal{J}'_{O_T}(s) + C_{O_T}(s)) \dot{s}^2$$

$$\mathbf{G}_{CT}(s) (-\mathbf{F}_{CT}(s)) + \mathbf{G}_M(s) (\mathbf{F}_M(s)) + {}^{O_T} \mathbf{f}_{T,ext}(s) = {}^{O_T} \mathbf{f}_{T,net}(s)$$

Where ${}^{O_C} \mathbf{f}_{C,net}$ and ${}^{O_C} \mathbf{f}_{C,ext}$ are the net wrench and the

external wrench acting on the cube expressed in the reference frame of the cube ($\{O_C\}$) and ${}^{O_T} \mathbf{f}_{T,net}$ and ${}^{O_T} \mathbf{f}_{T,ext}$ are the net wrench and external wrench acting on the tray expressed in the reference frame of the tray ($\{O_T\}$). By adding these equality constraints and the friction cone constraints at all the contacts to the time optimal control (32), we can formulate the non-prehensile manipulation task as a SOCP. For this experiment, we modified the intermediate pose of the tray such that the angle the tray makes with the horizontal increases between different trials (Fig 6). We use tilt angles of 0° , 5° , 10° and 15° . The coefficient of friction at the three tray-cube contacts was set as $\mu_1 = \mu_2 = \mu_3 = 0.275$. The other parameters were set as $\mu_M = 0.4$, $e_{M_x} = e_{M_y} = 1$, $e_{M_z} = 0.25$, $F_z = 50$ N, $m_{tray} = 0.125$ kg, $m_{cube} = 1$ kg, $K = 250$. The plots in Fig 5 show the friction cone constraints at the tray-cube contacts when the tilt angle is 10° (first three from left) and when the tilt angle is 15° . When the tilt angle, for the intermediate pose, was set to 16° , the optimization problem becomes infeasible, as the tray-cube contact constraints cannot be satisfied. This is evident from the fact that the contact constraints are at the limit for a tilt angle of 15° . We observed an optimal trajectory execution time of 4.14 seconds for a 0° tilt angle up to 5.87 seconds for a 15° tilt angle. For these trials, YALMIP took approximately 1.6 seconds on average to get the output after it is given the modeled problem.

VII. CONCLUSION

In this paper we have shown that the problem of time optimal motion generation considering the entire dynamics of the system and the full nonlinear friction cone constraints at the contacts can be formulated as a SOCP. Our formulation is general enough to consider v manipulators, each with one or more contacts with the manipulated object, as well as contacts of the object with the environment, if present. We also show that within the same framework, we can formulate non-prehensile manipulation tasks while satisfying contact constraints. The solution to our SOCP gives the time parameterization of the motion, joint torques of the manipulators, and also the contact forces at the manipulator-object and object-environment contacts. We evaluate our formulation on tasks like pivoting, where we have constraints on the motion through environmental contact, as well as tasks where we have constraints on the motion without environmental contact. In the future, we plan to study the effect of noise and model uncertainties on our proposed approach. We also aim to perform real-world experiments with multiple manipulators to investigate how the increase in computational load affects the time taken to solve the optimization problem.

REFERENCES

- [1] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing*, vol. 33, no. 2, pp. 387–424, 2022.
- [2] R. Laha, R. Sun, W. Wu, D. Mahalingam, N. Chakraborty, L. F. Figueredo, and S. Haddadin, "Coordinate invariant user-guided constrained path planning with reactive rapidly expanding plane-oriented escaping trees," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 977–984.
- [3] R. Laha, M. Becker, J. Vorndamme, J. Vrabel, L. F. Figueredo, M. A. Müller, and S. Haddadin, "Predictive multi-agent based planning and landing controller for reactive dual-arm manipulation," *IEEE Transactions on Robotics*, 2023.
- [4] C. Lin, P. Chang, and J. Luh, "Formulation and optimization of cubic polynomial joint trajectories for industrial robots," *IEEE Transactions on automatic control*, vol. 28, no. 12, pp. 1066–1074, 1983.
- [5] J. E. Bobrow, S. Dubowsky, and J. S. Gibson, "Time-optimal control of robotic manipulators along specified paths," *The international journal of robotics research*, vol. 4, no. 3, pp. 3–17, 1985.
- [6] Z. Shiller and S. Dubowsky, "Robot path planning with obstacles, actuator, gripper, and payload constraints," *IJRR*, 1989.
- [7] M. Suomalainen, Y. Karayiannidis, and V. Kyrki, "A survey of robot manipulation in contact," *Robotics and Autonomous Systems*, 2022.
- [8] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Practical time-optimal trajectory planning for robots: a convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 14, p. 28, 2008.
- [9] R. Laha, W. Wu, R. Sun, N. Mansfeld, L. F. Figueredo, and S. Haddadin, "S*: On safe and time efficient robot motion planning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 12758–12764.
- [10] I. Kao, K. M. Lynch, and J. W. Burdick, "Contact modeling and manipulation," *Springer Handbook of Robotics*, pp. 931–954, 2016.
- [11] A. Patankar, A. Fakhari, and N. Chakraborty, "Hand-object contact force synthesis for manipulating objects by exploiting environment," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9182–9189.
- [12] A. Fakhari, A. Patankar, and N. Chakraborty, "Motion and force planning for manipulating heavy objects by pivoting," in *2021 IEEE/RSJ Inter. Conf. on Intelligent Robots and Systems (IROS)*.
- [13] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.
- [14] M. Selvaggio, A. Garg, F. Ruggiero, G. Oriolo, and B. Siciliano, "Non-prehensile object transportation via model predictive non-sliding manipulation control," *IEEE Trans. Control Syst. Technol.*, 2023.
- [15] M. E. Kahn, *The Near-minimum-time Control of Open-loop Articulated Kinematic Chains*. Stanford University, 1969, no. 106.
- [16] M. Kahn, "The near-minimum-time control of open-loop articulated kinematic chains," *Trans. ASME, J. of Dyn. Sys. Meas. and Contr.*, vol. 93, pp. 164–172, 1971.
- [17] M. Niv and D. M. Auslander, "Optimal control of a robot with obstacles," in *1984 American Control Conference*. IEEE, 1984.
- [18] J. M. Hollerbach, "Dynamic scaling of manipulator trajectories," in *1983 American Control Conference*. IEEE, 1983, pp. 752–756.
- [19] K. Shin and N. McKay, "Minimum-time control of robotic manipulators with geometric path constraints," *IEEE Transactions on Automatic Control*, vol. 30, no. 6, pp. 531–541, 1985.
- [20] T. Kunz and M. Stilman, "Time-optimal trajectory generation for path following with bounded acceleration and velocity," *Robotics: Science and Systems VIII*, p. 209, 2013.
- [21] Q.-C. Pham, "A general, fast, and robust implementation of the time-optimal path parameterization algorithm," *IEEE Transactions on Robotics*, vol. 30, no. 6, pp. 1533–1540, 2014.
- [22] F. Lange and A. Albu-Schäffer, "Path-accurate online trajectory generation for jerk-limited industrial robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 82–89, 2015.
- [23] D. Verscheure, B. Demeulenaere, J. Swevers, J. De Schutter, and M. Diehl, "Time-optimal path tracking for robots: A convex optimization approach," *IEEE Transactions on Automatic Control*, vol. 54, no. 10, pp. 2318–2327, 2009.
- [24] T. Kröger and F. M. Wahl, "Online trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Transactions on Robotics*, vol. 26, no. 1, pp. 94–111, 2009.
- [25] S. He, C. Hu, S. Lin, and Y. Zhu, "An online time-optimal trajectory planning method for constrained multi-axis trajectory with guaranteed feasibility," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7375–7382, 2022.
- [26] M. S. Lobo, L. Vandenberghe, S. Boyd, and H. Lebret, "Applications of second-order cone programming," *Linear algebra and its applications*, vol. 284, no. 1–3, pp. 193–228, 1998.
- [27] S. P. Boyd and B. Wegbreit, "Fast computation of optimal contact forces," *IEEE Transactions on Robotics*, vol. 23, no. 6, pp. 1117–1132, 2007.
- [28] J.-P. Sleiman, J. Carius, R. Grandia, M. Wermelinger, and M. Hutter, "Contact-implicit trajectory optimization for dynamic object manipulation," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6814–6821.
- [29] A. O. Onol, P. Long, and T. Padir, "A comparative analysis of contact models in trajectory optimization for manipulation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1–9.
- [30] A. Ö. Onol, P. Long, and T. Padir, "Contact-implicit trajectory optimization based on a variable smooth contact model and successive convexification," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2447–2453.
- [31] A. Ö. Onol, R. Corcodel, P. Long, and T. Padir, "Tuning-free contact-implicit trajectory optimization," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020.
- [32] R. M. Murray, Z. Li, and S. S. Sastry, *A mathematical introduction to robotic manipulation*. CRC press, 2017.
- [33] S. P. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.
- [34] J. Löfberg, "Yalmip : A toolbox for modeling and optimization in matlab," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan.
- [35] J. F. Sturm, "Using sedumi 1.02, a MATLAB toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1–4, pp. 625–653, 1999. [Online]. Available: <https://doi.org/10.1080/10556789908805766>
- [36] F. Emika, "Franka emika panda hardware specifications," https://frankaemika.github.io/docs/control_parameters.html#limits-for-panda, accessed: March 2024.
- [37] A. Sarker, A. Sinha, and N. Chakraborty, "On screw linear interpolation for point-to-point path planning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [38] F. G. Flores and A. Kecskeméthy, "Time-optimal path planning for the general waiter motion problem," in *Advances in Mechanisms, Robotics and Design Education and Research*. Springer, 2013, pp. 189–203.
- [39] A. Heins and A. P. Schoellig, "Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator," *IEEE Robot. Autom. Lett.*, 2023.
- [40] R. I. C. Muchacho, R. Laha, L. F. Figueredo, and S. Haddadin, "A solution to slosh-free robot trajectory optimization," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 223–230.
- [41] H. Gattringer, A. Müller, S. Weitzhofer, and M. Schörgenhuber, "Point to point time optimal handling of unmounted rigid objects and liquid-filled containers," *Mechanism and Machine Theory*, vol. 184, p. 105286, 2023.