

# Hierarchical Reactive Grasping via Task-Space Velocity Fields and Joint-Space Quadratic Programming

Yonghyeon Lee<sup>1\*</sup>, Tzu-Yuan Lin<sup>1\*</sup>, Alexander Alexiev<sup>1</sup>, and Sangbae Kim<sup>1</sup>

**Abstract**—We present a fast and reactive grasping framework that combines task-space velocity fields with joint-space Quadratic Program (QP) in a hierarchical structure. Reactive, collision-free global motion planning is particularly challenging for high-DoF systems, as simultaneous increases in state dimensionality and planning horizon trigger a combinatorial explosion of the search space, making real-time planning intractable. To address this, we plan *globally* in a lower-dimensional task space – such as fingertip positions – and track *locally* in the full joint space while enforcing all constraints. This approach is realized by constructing velocity fields in multiple task-space coordinates (or, in some cases, a subset of joint coordinates) and solving a weighted joint-space QP to compute joint velocities that track these fields with appropriately assigned priorities. Through simulation experiments and real-world tests using the recent pose-tracking algorithm FoundationPose [1], we verify that our method enables high-DoF arm–hand systems to perform real-time, collision-free reaching motions while adapting to dynamic environments and external disturbances. Project page: <https://reactivegrasp.github.io>.

## I. INTRODUCTION

As robotic systems move toward unstructured and dynamic environments, the ability to adapt in real-time – referred to as *reactive control* – has become a key capability. Among core manipulation skills, grasping is one of the most fundamental and widely studied. Nonetheless, *reactive grasping* remains a challenging problem, particularly for high-dimensional systems, e.g., with multi-DoF grippers.

While recent work [2], [3] has advanced real-time grasp-pose generation and tracking, the complementary challenge of collision-free reactive motion planning has received less attention. RL-based approaches [4], [5], [6] are often restricted to single-object, obstacle-free, top-down reaches, which limits their generalizability to out-of-distribution scenarios. In contrast, we propose a model-based real-time planning and control framework that generalizes naturally across clutter and diverse grasping directions.

Reactive grasping requires both real-time perception and motion planning, and we focus on the latter. We assume the availability of perception modules capable of recognizing

object geometries and target fingertip positions, as well as dynamically tracking object poses – assumptions that are becoming increasingly realistic thanks to recent advances in deep visual perception models [1], [7], [8], [9], [10]. In particular, we address the reaching phase – the pre-grasp motion leading up to gripper closure – formulated as a collision-free motion generation problem. Even with ideal perception modules, the high dimensionality of the arm–hand system, coupled with the complex and often highly concave geometries of everyday objects, makes real-time feedback motion generation a fundamentally challenging problem.

A substantial body of work addresses collision-free motion generation with closed-loop reactivity, relying on Artificial Potential Fields (APF) [11], [12], [13], [14] and, more recently, on Dynamical Systems (DS) [15], [16], [17], [18], [19], where local modulation of a nominal motion plan enables efficient obstacle avoidance through analytical formulations. Recent variants provide formal convergence guarantees for certain obstacle classes, including convex shapes, star-shaped objects, and tree-of-stars geometries.

For reactive grasping, these methods can be applied in either task space (e.g., at the level of individual links) or joint space, but both present challenges. In task space, multiple potential fields or dynamical systems are usually combined by weighting control forces projected into joint space via the Jacobian. This often results in (i) tedious weight tuning, (ii) poor representation of highly concave obstacles (e.g., wine glasses) when using structured shape abstractions such as tree-of-stars, and (iii) combined fields that lack convergence guarantees and are prone to local minima. In joint space, collision constraints are typically highly non-convex and unstructured, making them difficult to approximate with compact geometric abstractions and limiting the applicability of reactive modulation approaches.

Iterative motion optimization methods, such as Model Predictive Control (MPC), can generate joint-space, collision-free motions in a closed-loop manner and, with sufficiently long horizons, can escape local minima to achieve global planning [21], [22], [23]. While traditionally too slow for global reactive planning with 7-DoF manipulators, recent advances – such as GPU-accelerated solvers, geometric planners, and optimized multi-threaded CUDA implementations – have significantly improved their practicality [24], [25]. Still, scaling to high-DoF arm–hand systems – often more than twice the dimensionality of the arm alone – remains difficult, as computational cost grows cubically with both state dimensionality and planning horizon. As a result, these methods are generally limited to quasi-reactive scenarios and

\*Equal contribution.

<sup>1</sup>The authors are with the Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (corresponding author: Yonghyeon Lee, yhl@mit.edu).

This work was supported in part by the Technology Innovation Program (RS-2024-00427719, Dexterous and Agile Humanoid Robots for Industrial Applications) funded by the Ministry of Trade, Industry and Energy (MOTIE, Korea), in part by the KIAT Global Industry Technology Cooperation Center Program (Project No. P246800183, Enhancing Robotic Manipulation Performance through Reflex-Based AI and AI-Compatible Hardware Design), and in part by the Advanced Robotics Lab of LG Electronics Co., Ltd.

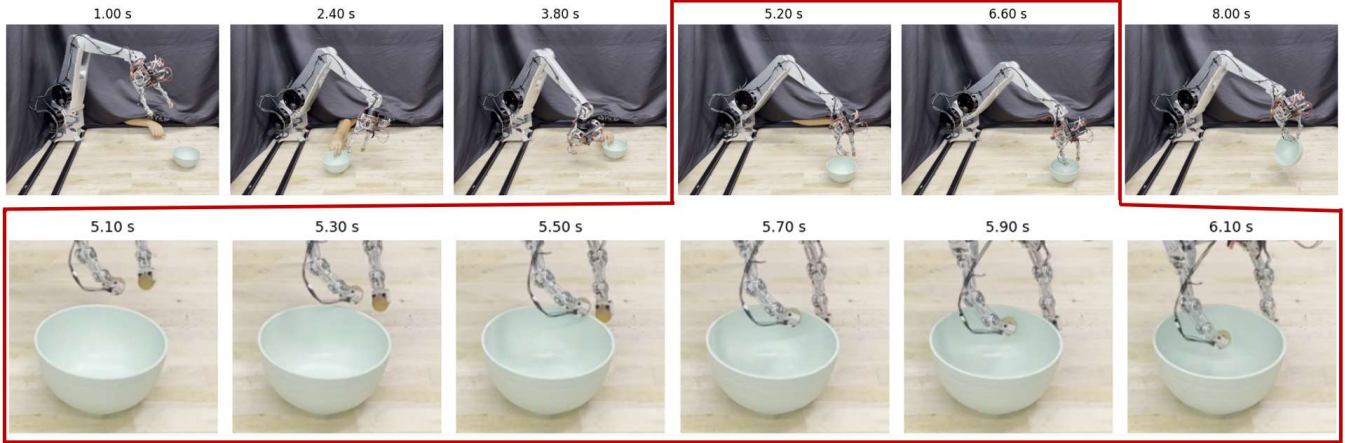


Fig. 1. *Reactive grasping* with a two-finger multi-DoF gripper (15-DoF arm–hand system: 7-DoF arm and 8-DoF hand; manipulation platform adapted from [20]). *Upper*: Under dynamic disturbances to the target bowl position, the robot reactively approaches to grasp it. *Lower*: Near the bowl, the fingers curl inward to avoid collisions with the concave object and converge to a stable grasp pose.

require problem-specific sampling strategies or strong warm-start heuristics [26], [27], [28], [29], [30].

In this paper, we present a highly reactive grasping framework for high-dimensional arm–hand systems that avoids the coupled growth of state and time-horizon dimensionality. The key idea is to plan *globally* in a lower-dimensional task space – keeping the state dimension fixed and low – with a subset of constraints, and to track *locally* in full joint space – keeping the time horizon fixed and short – while considering all constraints. This is implemented as a two-layer hierarchical control framework.

In the higher layer, we construct time-invariant velocity fields – first-order autonomous dynamical systems – in multiple task-space coordinates, such as fingertip positions and orientations, as well as in subsets of joint coordinates, such as the gripper joints. These fields provide high-level velocity targets that guide the robot toward a desired grasp pose without accounting for complex joint-space constraints. In the lower layer, we solve a joint-space Quadratic Program (QP) that incorporates all relevant constraints – including full-body collisions and joint velocity limits – to compute feasible joint velocities that track these high-level targets with appropriate weighting.

At the heart of our framework is the principle that the high-level velocity targets are informed by the *global* paths, enabling them to guide the robot toward the target grasp pose. To this end, we construct linear fingertip velocities by solving lightweight global 3D path optimizations and extracting their initial velocity directions – akin to the spirit of MPC, where only the first action is executed. As a complementary design choice, we also incorporate auxiliary analytic velocity fields for fingertip orientations and gripper joints to further enhance grasp stability.

This hierarchical structure decouples global guidance from local feasibility: global planning operates in a fixed low-dimensional space with limited constraints, while local tracking is enforced through a one-step joint-space QP with full constraints, yielding a reactive strategy that is globally informed yet locally feasible.

We validate our framework through experiments in both simulation and the real world using a 15-DoF arm–hand system (7-DoF arm and 8-DoF hand; manipulation platform adapted from [20]), demonstrating real-time collision-free reaching motion planning around concave target objects and multiple obstacles, as well as robust reactive grasping under disturbances to the robot and target objects; for example, Fig. 1 presents snapshots from real-world experiments.

## II. METHODS

We begin this section by introducing the assumptions and notations used throughout the paper. We consider a robotic manipulator equipped with a multi-DoF gripper comprising  $m$  fingertips. The full arm–hand system has  $n$  degrees of freedom and is fully actuated. The joint configuration is denoted by  $q \in \mathbb{R}^n$ , while the position and orientation of the  $i$ -th fingertip are denoted by  $x_i \in \mathbb{R}^3$  and  $R_i \in \mathbb{R}^{3 \times 3}$  for  $i = 1, \dots, m$ , respectively. We denote the concatenated vector of fingertip positions by  $\mathbf{x} = (x_1, \dots, x_m) \in \mathbb{R}^{3m}$ . The Jacobians associated with the  $i$ -th fingertip’s position and orientation are denoted by  $J_i^x(q) \in \mathbb{R}^{3 \times n}$  and  $J_i^R(q) \in \mathbb{R}^{3 \times n}$ , respectively, such that

$$\dot{x}_i = J_i^x(q)\dot{q}, \quad \dot{R}_i R_i^\top = [J_i^R(q)\dot{q}], \quad (1)$$

where  $[\omega] \in \mathbb{R}^{3 \times 3}$  denotes the skew-symmetric matrix corresponding to the angular velocity vector  $\omega \in \mathbb{R}^3$ , such that  $[\omega]v = \omega \times v$  for any  $v \in \mathbb{R}^3$  [31].

We denote by  $\Gamma(q) \in \mathbb{R}^k$  the vector of minimum Euclidean distances at configuration  $q$  between each rigid link of the robot and the fixed environment (e.g., table, fixture, wall), as well as self-collision distances, where  $k$  is the total number of considered collision pairs. Ideally,  $\Gamma(q)$  is differentiable everywhere; in practice, we assume that it is continuous and differentiable almost everywhere, and that the robot rarely passes through the non-differentiable points.

We assume that the geometry of the target object to grasp  $\mathcal{O}$  is represented by a signed distance function  $d(x, \mathcal{O})$ , which outputs the signed Euclidean distance from a point  $x$  to the surface of  $\mathcal{O}$  – negative inside the object, zero

on the surface, and positive outside [32]. The object is allowed to move dynamically, and a perception module is assumed to provide its time-varying signed distance function in real time, for example by tracking the object's pose. With obstacles present,  $d(x, \mathcal{O})$  can incorporate their signed distances, becoming vector-valued.

In addition, we assume that the perception module provides, in real time, a set of candidate fingertip positions

$$\{\mathbf{x}_j^* \in \mathbb{R}^{3m}\}_{j=1}^N \quad \text{s.t.} \quad \mathbf{x}_j^* = (x_{j,1}^*, \dots, x_{j,m}^*),$$

for the pre-grasp reaching phase (see Fig. 4 for example). When these targets are predefined in the object frame, real-time generation reduces to tracking the object pose.

Given  $d(x, \mathcal{O})$  and  $\mathbf{x}_j^*$  in real time, we perform online reactive motion planning, executed at rates exceeding 50 Hz to ensure reactivity. At each planning iteration, we select the target fingertip positions  $\mathbf{x}^* \in \mathbb{R}^{3m}$  from the candidate set as those closest to the current positions  $\mathbf{x}$ , according to a suitable distance metric (this metric should be tailored to the specific gripper, and we provide an example in the experimental section). Then, our closed-loop planning algorithm for computing the desired joint velocities  $\dot{q}_{\text{des}} \in \mathbb{R}^n$  consists of the following three steps.

First, we solve a 3D collision-free path optimization for each fingertip, using the current positions  $\mathbf{x}$  and selected targets  $\mathbf{x}^*$  as boundary conditions. The optimization considers a subset of constraints, including the signed distances  $d(x, \mathcal{O})$  to the target and non-target objects, as well as collisions with the table. These global paths define local velocity fields for the fingertip positions; details are provided in Section II-A. Next, we construct auxiliary velocity fields for fingertip orientations and gripper joints to enhance grasp stability; see Section II-B. Finally, the combined velocity fields are locally tracked by solving a joint-space Quadratic Program (QP) that computes feasible joint velocities under full-body collision and joint limit constraints; see Section II-C. Implementation details are provided in Section II-D.

### A. Fingertip Velocity Fields via 3D Path Optimization

A straightforward approach to constructing a velocity field that drives the fingertips toward the target positions  $\mathbf{x}^*$  is to use a linear attractor of the form

$$\dot{\mathbf{x}}_{\text{des}} = K(\mathbf{x}^* - \mathbf{x}), \quad (2)$$

where  $K \in \mathbb{R}^{3m \times 3m}$  is a positive definite gain matrix, and  $\mathbf{x} = (x_1, \dots, x_m)$ ,  $\mathbf{x}^* = (x_1^*, \dots, x_m^*)$  denote the current and target fingertip positions, respectively [33]. However, when linear paths are obstructed by the target object itself – even in the case of simple shapes such as cubes – tracking this field under collision constraints can cause the robot to become trapped in local minima.

To provide global guidance and mitigate local minima, we define a velocity field for each fingertip by solving a global path optimization problem in  $\mathbb{R}^3$ . Specifically, for the  $i$ -th

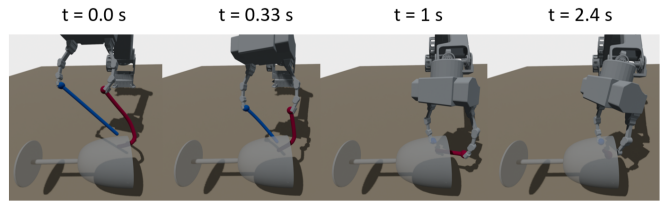


Fig. 2. Real-time 3D path optimization for the fingertips.

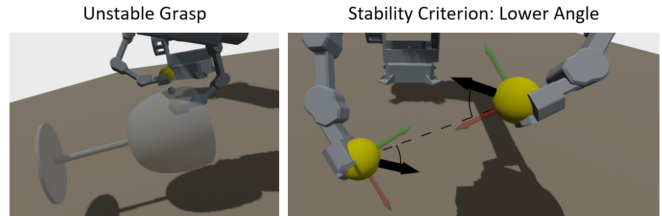


Fig. 3. *Left*: An example of an unstable gripper configuration. *Right*: Stability score given by the angle between dotted line and black arrow.

fingertip position  $x_i$ , we solve

$$\min_{c_i(s)} \int_0^1 \left\| \frac{dc_i(s)}{ds} \right\|^2 ds \quad \text{s.t.} \quad d(c_i(s), \mathcal{O}) \geq \epsilon_{\mathcal{O}} \\ c_i(0) = x_i, \quad c_i(1) = x_i^* \quad (3)$$

where  $c_i(s)$  is a continuous path parameterized by  $s \in [0, 1]$ ,  $\mathcal{O}$  is the target object, and  $\epsilon_{\mathcal{O}} > 0$  is a safety margin to ensure clearance from the object surface. When there are other non-target objects or obstacles in the scene (e.g., tissue boxes in Fig. 11), they can be incorporated as additional constraints using signed distance functions, with their geometry approximated by simple convex shapes that encapsulate them, such as superellipsoids [7], since precise geometry is unnecessary. A relatively larger safety margin (e.g., 5 cm) can be used for non-target objects, whereas a tighter margin (e.g., 1 cm) is required for the target object. This 3D path optimization can be solved in real time for moderately complex objects using modern solvers [34], [35]; Fig. 2 shows snapshots of the real-time fingertip path optimization results.

Let  $x_i(s)$  be the solution of (3), we then extract the initial velocity vector  $\dot{c}_0 = \frac{dx_i}{ds}(0)$  from the optimized path and define the desired fingertip velocity as

$$\dot{x}_{i,\text{des}} := v(x_i, x_i^*) \frac{\dot{c}_0}{\|\dot{c}_0\|}, \quad (4)$$

where  $v(x_i, x_i^*) \in \mathbb{R}$  is a scalar speed function that depends on the current and target fingertip positions. To ensure a constant speed when the fingertip is far from the target and a smooth, quadratic decay to zero as it approaches the target, we define  $v(x_i, x_i^*)$  as follows. Let  $e_i := \|x_i - x_i^*\|$  denote the distance to the target. Then

$$v(x_i, x_i^*) := \begin{cases} v_{\text{const}}, & \text{if } e_i > \delta_x, \\ v_{\text{const}} \left( 1 - \left( 1 - \frac{e_i}{\delta_x} \right)^2 \right), & \text{otherwise,} \end{cases} \quad (5)$$

where  $v_{\text{const}}$  is a constant maximum speed and  $\delta_x$  is a threshold distance for speed modulation.

### B. Auxiliary Velocity Fields for Grasp Stability

Tracking only the linear fingertip velocities can drive the fingertips to their target positions, but may result in orientations that are unsuitable for stable grasping. For instance, as illustrated in Fig. 3 (*Left*), in a two-finger gripper where each fingertip has a designated contact surface (yellow surface), the unintended part of the fingertip – rather than the intended soft contact region – may come into contact with the object, resulting in unstable or undesired grasps.

We assume that the stable grasping criterion is given as a scalar function of the fingertip positions and orientations,

$$g(x_1, R_1, \dots, x_m, R_m) \in \mathbb{R}, \quad (6)$$

where the grasp is considered stable if  $g \leq \eta_g$ , with  $\eta_g > 0$  being a predefined stability threshold, and unstable otherwise. For example, in the case of a two-finger gripper shown in Fig. 3 (*Right*), the angle between the fingertip difference vector  $x_2 - x_1$  (or  $x_1 - x_2$ ) and the central axis of the desired contact region (illustrated as black arrows) – computed from the fingertip orientation matrix  $R_1$  (or  $R_2$ ) – can be used as the stability score  $g$ .

To guide the fingertip orientations toward satisfying the stable grasping criterion, we define an auxiliary angular velocity field for each fingertip as

$$\omega_{i,\text{des}} = -\text{unskew} \left( \frac{\partial g}{\partial R_i} R_i^\top \right), \quad (7)$$

where  $\text{unskew}(\cdot) \in \mathbb{R}^3$  maps a  $3 \times 3$  skew-symmetric matrix to its corresponding vector in  $\mathbb{R}^3$  (i.e., the inverse of the skew operator  $[\cdot]$ ) and  $\frac{\partial g}{\partial R_i} \in \mathbb{R}^{3 \times 3}$  is a  $3 \times 3$  gradient matrix.

Tracking the desired angular velocities in (7) becomes less important when the stability criterion is already satisfied, i.e.,  $g \leq \eta_g$ , and should therefore be given lower priority in the tracking stage. To account for this, we introduce a weight parameter that depends on  $g$ :

$$\alpha(g) := \frac{1}{2} + \frac{1}{2} \tanh \left( \frac{g - \eta_g}{T_g} \right), \quad (8)$$

where  $T_g$  is a temperature parameter that controls the smoothness of the transition. This function smoothly increases from 0 to 1 as  $g$  increases past  $\eta_g$ , emphasizing orientation correction only when necessary.

Lastly, we introduce an auxiliary velocity field for the gripper posture, which drives the gripper toward a nominal configuration – loosely open and ready to grasp – when the current fingertip positions are far from the target. This nominal posture can often be easily predefined by the user. Specifically, let  $q_{\text{gr}}$  denote the current gripper joint value and  $q_{\text{gr}}^*$  the nominal gripper joint configuration. We define the desired gripper joint velocity as

$$\dot{q}_{\text{gr},\text{des}} := v(q_{\text{gr}}, q_{\text{gr}}^*) \frac{q_{\text{gr}}^* - q_{\text{gr}}}{\|q_{\text{gr}}^* - q_{\text{gr}}\|}, \quad (9)$$

where  $v(q_{\text{gr}}, q_{\text{gr}}^*)$  is a scalar speed function, defined analogously to (5) with a threshold parameter  $\delta_{\text{gr}}$ .

As with the fingertip angular velocity fields, this term should be downweighted when the fingertips are already

close to their target positions. Denoting the fingertip position error by  $h(\mathbf{x}, \mathbf{x}^*) = \|\mathbf{x} - \mathbf{x}^*\|$ , we define a weight parameter:

$$\beta(h) := \frac{1}{2} + \frac{1}{2} \tanh \left( \frac{h - \eta_h}{T_h} \right), \quad (10)$$

where  $\eta_h$  is a threshold distance and  $T_h$  controls the smoothness of the transition.

### C. Weighted Velocity Field Tracking via QP

In this section, we present a QP formulation for tracking the desired velocity fields – including the fingertip linear velocities  $\dot{x}_{i,\text{des}}$  (4), the fingertip angular velocities  $\omega_{i,\text{des}}$  (7) weighted by  $\alpha(g)$  (8), and the gripper joint velocity  $\dot{q}_{\text{gr},\text{des}}$  (9) weighted by  $\beta(h)$  (10) – subject to full collision constraints, including  $\Gamma(q)$  for robot-environment and self-collisions, and  $d(x, \mathcal{O})$  for target object and obstacles avoidance.

We consider a constant desired joint velocity  $\dot{q}_{\text{des}}$  over a planning horizon  $H > 0$  – which is a real scalar value –, and approximate the future joint configuration as  $q + \dot{q}_{\text{des}} H$ . Drawing inspiration from [36], [37], where a quadratic program is employed to solve inverse kinematics in real time, we formulate a QP that tracks the desired velocities by linearizing all constraints around the current configuration  $q$ . A distinctive feature of our approach is the tracking of weighted velocity fields using position-dependent weights  $\alpha$  and  $\beta$ :

$$\begin{aligned} \dot{q}_{\text{des}} = \arg \min_{\dot{q}} & \sum_{i=1}^m \|\dot{x}_{i,\text{des}} - J_i^x(q)\dot{q}\|^2 \\ & + \alpha(g) \|\omega_{i,\text{des}} - J_i^R(q)\dot{q}\|^2 \\ & + \beta(h) \|\dot{q}_{\text{gr},\text{des}} - S\dot{q}\|^2 \\ \text{subject to} & \Gamma(q) + \frac{d\Gamma(q)}{dq} \dot{q} H \geq \epsilon_\Gamma, \\ & d(X(q), \mathcal{O}) + \frac{\partial d(X(q), \mathcal{O})}{\partial q} \dot{q} H \geq \epsilon_\mathcal{O}, \\ & q_{\text{max}} \geq q + \dot{q} H \geq q_{\text{min}}, \\ & \dot{q}_{\text{max}} \geq \dot{q} \geq \dot{q}_{\text{min}}, \end{aligned} \quad (11)$$

where  $S \in \mathbb{R}^{n_{\text{gr}} \times n}$  is a selection matrix that extracts the gripper joint components from the full joint velocity vector  $\dot{q} \in \mathbb{R}^n$ , and  $n_{\text{gr}}$  is the number of gripper joints. The term  $\Gamma(q) \in \mathbb{R}^k$  encodes robot-environment and self-collision distances, with threshold  $\epsilon_\Gamma > 0$ . The function  $d(X, \mathcal{O}) \in \mathbb{R}^l$  represents the signed distances between a set of points  $X \subset \mathbb{R}^{3l}$ , approximating the gripper geometry, and the target object and non-target obstacles  $\mathcal{O}$ , with safety margin  $\epsilon_\mathcal{O} > 0$ . Lastly,  $q_{\text{max}}$  and  $q_{\text{min}}$  denote the joint position limits, and  $\dot{q}_{\text{max}}$  and  $\dot{q}_{\text{min}}$  denote the joint velocity limits.

The above QP can be solved efficiently using analytical expressions for the Jacobians and the gradients of  $\Gamma(q)$  and  $d(X, \mathcal{O})$ . The planning horizon  $H$  and the margin parameters  $\epsilon$  control the conservativeness of the tracking behavior. In narrow passages, it may be necessary to reduce the desired velocity magnitude and choose a smaller planning horizon  $H$ , along with tighter margins  $\epsilon$ .

#### D. Implementation Details

This section provides implementation details. The specifications described here can be adapted to suit the user’s specific application or system setup. We use Pinocchio [38] for computing forward kinematics and Jacobians, and FCL [39] for computing distances and nearest points between collision geometries. These tools are used to evaluate  $\Gamma(q)$  and its gradient. For convex objects, we employ superquadric models to represent the signed distance function [7]. For concave objects with thin surfaces (e.g., bowls, dishes, or the rim of a wine glass), we use a surface point cloud and define the signed distance as the nearest-point distance to this cloud. Given the thinness of the surface, ignoring the sign inside the object does not significantly affect the result. As this computation can be expensive, we accelerate it using GPU-based nearest neighbor queries.

For the 3D path optimization (3), we discretize  $c_i(s)$  using 100 uniformly spaced grid points and solve the problem using Sequential Quadratic Programming (SQP) with a small number of iterations (typically  $\approx 3$ ). For initialization, we use a GPU-accelerated sampling-based search to identify a single via point, positioned at a fixed distance (e.g., 10 cm) from the target, defining a piecewise linear path of two segments with minimal constraint violation. We use OSQP [34] as the underlying solver for both the SQP used in the 3D path optimization and the joint-space QP (11). Given  $\dot{q}_{des}$  from (11), we integrate over one control interval to obtain the desired joint position and then apply PD control with gravity compensation.

### III. EXPERIMENTS

We conduct case studies of the proposed reactive grasping method using a 15-DoF arm-hand manipulation system [20], consisting of a 7-DoF arm and two 4-DoF fingers (as shown in previous figures). We first describe several design choices in our framework that are specific to this platform. For the distance metric used to select the target fingertip positions – i.e., to compare  $\mathbf{x} = (x_1, x_2)$  and  $\mathbf{x}^* = (x_1^*, x_2^*)$  – we use

$$\|x_1 - x_1^*\| + \|x_2 - x_2^*\| + w \cdot \angle(x_2 - x_1, x_2^* - x_1^*), \quad (12)$$

with a positive weight  $w$ . The function  $g(x_1, x_2, R_1, R_2)$  for the stable grasping criterion in (6) is defined as the mean of the angles

$$\angle(x_2 - x_1, R_1^x + R_1^y) \quad \text{and} \quad \angle(x_1 - x_2, R_2^x + R_2^y),$$

where  $R^x$  and  $R^y$  denote the  $x$ - and  $y$ -axes of the orientation matrix, respectively, as illustrated in Fig. 3 (Right).

#### A. Simulation Studies

In this section, we present simulation studies in the Mujoco [40] physics engine using a diverse set of objects, as shown in Fig. 4. Each object is assumed to have predefined target fingertip positions, indicated by pairs of red and blue points; example collision geometries are shown in Fig. 4 (Right). The maximum planning frequency for  $\dot{q}_{des}$  is set to 100 Hz, running in a separate thread from the main control loop. Once the fingertips come within a specified threshold

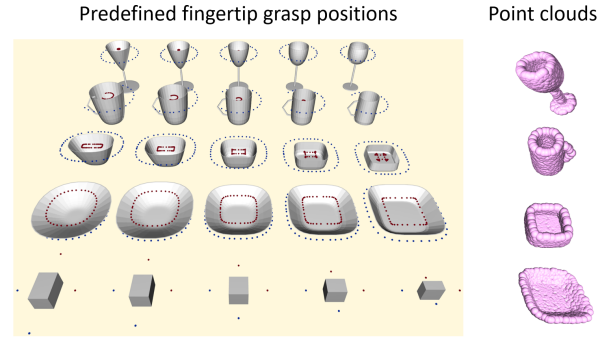


Fig. 4. Left: A diverse set of objects, whose shapes and target fingertip positions are adopted from T<sup>2</sup>SQNet [8]. Right: Given point cloud representations of concave object surfaces, a distance threshold of  $\epsilon_O = 1$  cm yields collision geometries modeled as a set of spheres.

TABLE I

GRASP SUCCESS RATES ACROSS VARYING SHAPES, SCALES, AND INITIAL CONDITIONS.

Method	Box	Bowl	Dish	Mug	Wine Glass
Linear	100.0%	33.3%	93.3%	30.7%	32.0%
Ours	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>	<b>100.0%</b>

(1 cm) of the target positions, the gripper closes, the normal force is gradually increased until the object is firmly secured without slipping, and the object is lifted.

First, we demonstrate our algorithm’s ability to avoid local minima and successfully reach target positions for concave objects, in contrast to a simple linear fingertip velocity field in (2) adopted from [33]. Table I reports the grasp success rates. The evaluation was conducted over three initial robot poses: gripper at left-bottom, center-up, and right-bottom. For each pose, we tested five objects in each of five categories. The Mug and Wine Glass were evaluated under five orientations (upright and rotated 90 degrees about each horizontal axis), while the other objects were tested only in the upright orientation. An attempt is considered successful if the robot lifts the object within 20 seconds. The results clearly demonstrate the superiority of our method. Example successful grasping trajectories are visualized in Fig. 5.

The causes of failure for the linear methods can be seen in Fig. 6, which shows the fingertip position errors at convergence (i.e., at terminal time). As shown, the linear methods are frequently trapped in local minima, resulting in consistently large fingertip errors for all concave shapes. In particular, a closer look at the results for the Dish in Fig. 6 is interesting. For our method, the Dish exhibits slightly larger errors (up to 2 cm) compared to other objects (mostly below 1 cm). This is due to its thin shape, which brings a fingertip close to the table, where collision constraints slightly disturb the motion during gripper closing. For the linear methods, performance on the Dish is relatively better than on other concave objects, because its concavity is less pronounced. Although the linear paths result in collisions, iterative replanning helps avoid them.

Next, we perform a perturbation analysis. Fig. 7 illustrates the system’s real-time response to external disturbances,



Fig. 5. Example successful grasping trajectories using our method; robot motion progresses from red to blue.

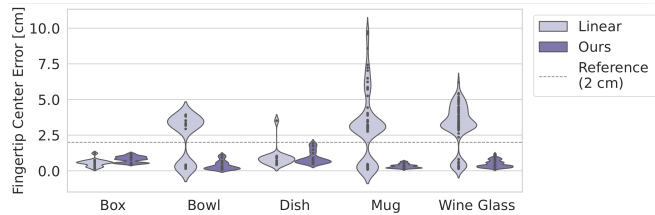


Fig. 6. Distributions of fingertip center position errors at convergence, which must fall below approximately 2 cm to be considered successful.

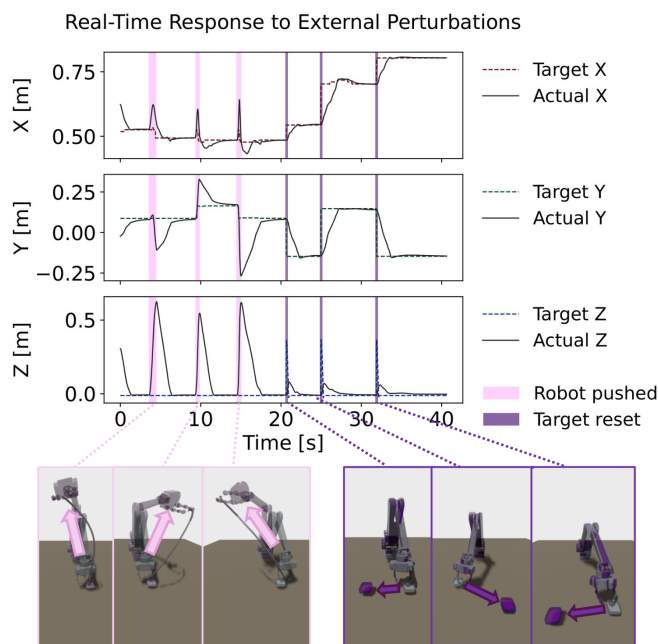


Fig. 7. Real-time response of our method to external perturbations. Shaded pink regions indicate physical pushes to the robot, and shaded purple regions mark sudden changes in the target object’s position. Dashed lines represent the target fingertip center coordinates  $(x, y, z)$ , while solid lines indicate the actual trajectories.

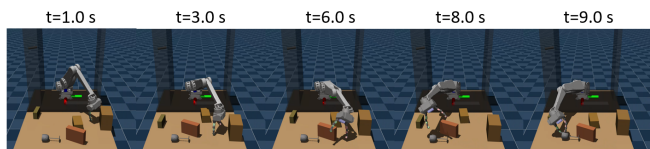


Fig. 8. Reactive reaching toward the wine glass without colliding with the moving boxes.

including physical pushes to the robot (marked in light pink) and sudden changes in the target object’s position (marked in dark purple), both triggered interactively in the physics engine. Dotted curves denote the target  $x$ -,  $y$ -, and  $z$ -coordinates of the fingertip center positions. While changes in the target position cause discontinuous jumps in these curves, slight changes also occur during robot pushes –

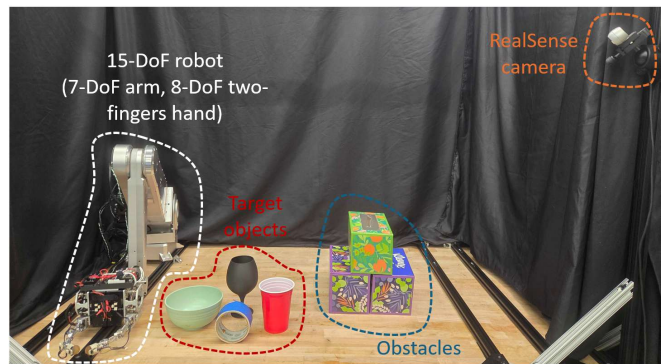


Fig. 9. Experimental setup.

even when the object remains stationary – due to updates in the nearest target point as the robot moves. Overall, the robot consistently recovers and re-converges to the target despite perturbations, demonstrating the real-time closed-loop reactivity of our grasping method.

Lastly, Fig. 8 shows our method applied to a scene with multiple non-target moving objects that limit the collision-free area; note that we use superellipsoids for these objects, while the target is represented by a point cloud. Our reactive grasping successfully navigates around the four moving boxes and reaches the target wine glass without collision.

### B. Real Robot Experiments

This section evaluates the proposed method in real-world settings, examining its ability to respond in real time to dynamic disturbances of the robot and target object, and to navigate through multiple obstacles. An overview of the experimental setup is shown in Fig. 9. We deploy the proposed algorithm on a 15-DoF robot manipulation system from [20]. In total, four concave objects and three obstacles of the same geometry are used in the experiments. For perceptual measurements, we use an externally mounted Intel RealSense D455 camera. In all experiments, we consider collisions with the walls, table, obstacles (modeled as superellipsoids [8]), target objects (represented as point clouds), and the robot itself.

Throughout the experiments, we assume prior knowledge of the target object model and its predefined grasp points. Object meshes are reconstructed with BundleSDF [9] and tracked with FoundationPose [1] for real-time pose estimation. Grasp fingertip positions are manually specified, as in Fig. 4 (Left). The robot is controlled at 500 Hz with a separate planning thread at 100 Hz (the actual planning speed can vary with factors such as the number of obstacles, but it generally remains above 50 Hz). As in the simulations, once the fingertips are within 1.5 cm of their targets, the gripper closes and lifts the object.

An episode is deemed successful if the robot lifts the object. Collisions with the target object are not considered failures, while collisions with other obstacles are (except incidental contacts with unmodeled elements such as cables). Trials were excluded if failures were clearly due to hardware issues – e.g., motor failures – or if FoundationPose [1] completely lost the object pose, primarily from occlusions, to

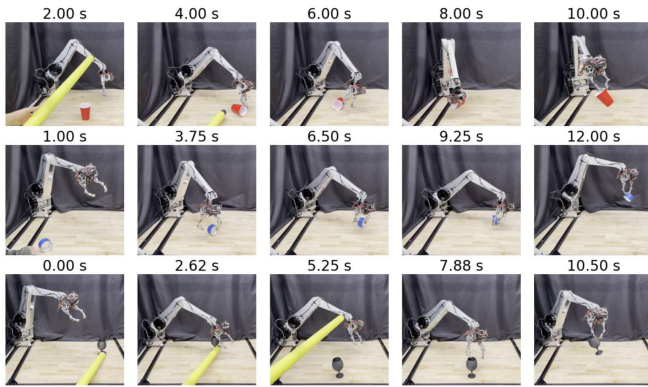


Fig. 10. Examples of strong disturbances (large deviations in object position and strong pushes to the robot).

TABLE II

GRASP SUCCESS RATES FOR THE BOWL WITH VARYING NUMBERS OF OBSTACLES (SUCCESSES OUT OF 10 TRIALS).

# Obstacles	1	2	3
	8/10	8/10	7/10

isolate algorithmic performance from mechanical limitations and perception failures.

First, we evaluate single-object grasping without additional obstacles, under strong disturbances applied either to the robot or to the object (see Fig. 10). These disturbances are introduced after the robot has begun moving and occur multiple times per trial. Specifically, while conducting strictly controlled experiments is challenging, we constrain the disturbances to remain within reasonable bounds: (i) the objects must stay within the robot’s workspace, (ii) they must not be moved into ungraspable configurations (e.g., an upside-down bowl), and (iii) their motion must remain within the speed range that FoundationPose can reliably track (see supplementary videos for examples). We performed five trials per object. The bowl, cup, and wine glass tasks were all successful, whereas the tape task resulted in one failure. Although the robot fingers occasionally collided with the target object due to tracking errors, the system was generally able to recover and complete the grasp. The failure with the tape object – where the gripper closed prematurely while the tape was still rolling, resulting in a missed grasp – was an expected outcome, as no prediction module for the object’s future position was used.

Second, we evaluate grasping performance in the presence of obstacles, as illustrated in Fig. 11. The target object and obstacles are randomly placed, and the reaching controller is executed without external disturbances. Table II reports the success rates over 10 trials for the cases with one, two, and three obstacles. Fig. 11 illustrates a successful case. The executed fingertip paths are shown with inter-fingertip segments (*Left*), alongside examples of planned fingertip paths (*Right*). From the right figure, two observations can be made: (i) the target grasp positions change as the motion progresses, and (ii) the planned and executed paths do not coincide exactly. The first observation is expected, since

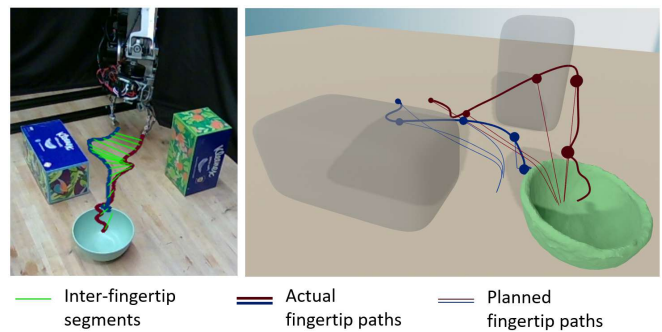


Fig. 11. An example of successful navigation around obstacles. *Left*: Visualization of fingertip paths (blue and red) and instantaneous inter-fingertip segments (green). *Right*: Thicker curves show the executed fingertip paths, while thinner curves indicate the real-time planned fingertip paths.



Fig. 12. Failure cases due to obstacles and walls.

grasp targets are updated online based on the current fingertip positions. The second arises because the low-level joint-space QP must enforce constraints and track additional velocity objectives at the same time, making exact tracking of the planned fingertip velocities inherently impossible. As shown in the figure, the planned fingertip paths nevertheless serve to continuously guide the fingertip motion toward the targets in real time, while avoiding collisions with the obstacles and the object.

Fig. 12 presents examples of failure cases, showing a limitation of our approach. In particular, the high-level velocity targets – especially for fingertip positions – are planned independently of the current full joint configuration. As a result, although the paths may appear collision-free from the fingertip’s perspective, following them can lead the robot into obstacle-facing configurations (*Left*) or local minima in narrow passages that it cannot traverse (*Right*).

#### IV. DISCUSSION AND CONCLUSION

We have proposed a hierarchical control framework for reactive grasping that is robust to external disturbances, where the higher layer defines task-space velocity targets and the lower layer tracks them under full joint-level constraints with appropriate weighting. The framework enables robust grasping of highly concave objects, such as wine glasses and bowls, even in the presence of obstacles.

Our framework can be improved in two directions. First, it relies on pre-defined grasp points, limiting adaptability to novel objects; this could be addressed by incorporating learning-based methods to predict grasp points online. Second, in cluttered environments with multiple obstacles, the system may become trapped in local minima (see Fig. 12). Adding velocity targets at the elbow or other joints, potentially through lower-frequency joint-level optimization as in [41], can be a promising solution.

## REFERENCES

- [1] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.
- [2] J. Liu, R. Zhang, H.-S. Fang, M. Gou, H. Fang, C. Wang, S. Xu, H. Yan, and C. Lu, "Target-referenced reactive grasping for dynamic objects," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 8824–8833.
- [3] H.-S. Fang, C. Wang, H. Fang, M. Gou, J. Liu, H. Yan, W. Liu, Y. Xie, and C. Lu, "Anygrasp: Robust and efficient grasp perception in spatial and temporal domains," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3929–3945, 2023.
- [4] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "Dexpoint: Generalizable point cloud reinforcement learning for sim-to-real dexterous manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 594–605.
- [5] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. Van Wyk, "Dextrahrgb: Visuomotor policies to grasp anything with dexterous hands," *arXiv preprint arXiv:2412.01791*, 2024.
- [6] B. Huang, J. Yu, and S. Jain, "Earl: Eye-on-hand reinforcement learner for dynamic grasping with active pose estimation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2963–2970.
- [7] S. Kim, T. Ahn, Y. Lee, J. Kim, M. Y. Wang, and F. C. Park, "Dsqnet: A deformable model-based supervised learning algorithm for grasping unknown occluded objects," *IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 3, pp. 1721–1734, 2022.
- [8] Y. H. Kim, S. Kim, Y. Lee, and F. C. Park, "T2sqnet: A recognition model for manipulating partially observed transparent tableware objects," in *8th Annual Conference on Robot Learning*, 2024.
- [9] B. Wen, J. Tremblay, V. Blukis, S. Tyree, T. Müller, A. Evans, D. Fox, J. Kautz, and S. Birchfield, "Bundlesdf: Neural 6-dof tracking and 3d reconstruction of unknown objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 606–617.
- [10] A. Millane, H. Oleynikova, E. Wirbel, R. Steiner, V. Ramasamy, D. Tingdahl, and R. Siegwart, "nvbox: Gpu-accelerated incremental signed distance field mapping," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 2698–2705.
- [11] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *The international journal of robotics research*, vol. 5, no. 1, pp. 90–98, 1986.
- [12] E. Rimon, *Exact robot navigation using artificial potential functions*. Yale University, 1990.
- [13] B. Lacevic, P. Rocco, and A. M. Zanchettin, "Safety assessment and control of robotic manipulators using danger field," *IEEE Transactions on Robotics*, vol. 29, no. 5, pp. 1257–1270, 2013.
- [14] S. Paternain, D. E. Koditschek, and A. Ribeiro, "Navigation functions for convex potentials in a space with convex obstacles," *IEEE Transactions on Automatic Control*, vol. 63, no. 9, pp. 2944–2959, 2017.
- [15] V. M. Goncalves, L. C. Pimenta, C. A. Maia, B. C. Dutra, and G. A. Pereira, "Vector fields for robot navigation along time-varying curves in  $n$ -dimensions," *IEEE Transactions on Robotics*, vol. 26, no. 4, pp. 647–659, 2010.
- [16] S. M. Khansari-Zadeh and A. Billard, "A dynamical system approach to realtime obstacle avoidance," *Autonomous Robots*, vol. 32, pp. 433–454, 2012.
- [17] L. Huber, J.-J. Slotine, and A. Billard, "Avoiding dense and dynamic obstacles in enclosed spaces: Application to moving in crowds," *IEEE Transactions on Robotics*, vol. 38, no. 5, pp. 3113–3132, 2022.
- [18] A. Billard, S. Mirrazavi, and N. Figueroa, *Learning for adaptive and reactive robot control: a dynamical systems approach*. MIT Press, 2022.
- [19] L. Huber, J.-J. Slotine, and A. Billard, "Avoidance of concave obstacles through rotation of nonlinear dynamics," *IEEE Transactions on Robotics*, vol. 40, pp. 1983–2002, 2023.
- [20] A. SaLoutos, H. Kim, E. Stanger-Jones, M. Guo, and S. Kim, "Towards robust autonomous grasping with reflexes using high-bandwidth sensing and actuation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 10 254–10 260.
- [21] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [22] G. Williams, N. Wagener, B. Goldfain, P. Drews, J. M. Rehg, B. Boots, and E. A. Theodorou, "Information theoretic mpc for model-based reinforcement learning," in *2017 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2017, pp. 1714–1721.
- [23] G. Williams, A. Aldrich, and E. A. Theodorou, "Model predictive path integral control: From theory to parallel computation," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 344–357, 2017.
- [24] M. Bhardwaj, B. Sundaralingam, A. Mousavian, N. D. Ratliff, D. Fox, F. Ramos, and B. Boots, "Storm: An integrated framework for fast joint-space model-predictive control for reactive manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 750–759.
- [25] B. Sundaralingam, S. K. S. Hari, A. Fishman, C. Garrett, K. Van Wyk, V. Blukis, A. Millane, H. Oleynikova, A. Handa, F. Ramos *et al.*, "Curobo: Parallelized collision-free robot motion generation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8112–8119.
- [26] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones, "Efficient interior point methods for multistage problems arising in receding horizon control," in *2012 IEEE 51st IEEE conference on decision and control (CDC)*. IEEE, 2012, pp. 668–674.
- [27] S. Richter, C. N. Jones, and M. Morari, "Computational complexity certification for real-time mpc with input constraints based on the fast gradient method," *IEEE Transactions on Automatic Control*, vol. 57, no. 6, pp. 1391–1403, 2011.
- [28] J. Schulman, Y. Duan, J. Ho, A. Lee, I. Awwal, H. Bradlow, J. Pan, S. Patil, K. Goldberg, and P. Abbeel, "Motion planning with sequential convex optimization and convex collision checking," *The International Journal of Robotics Research*, vol. 33, no. 9, pp. 1251–1270, 2014.
- [29] J. Sacks and B. Boots, "Learning to optimize in model predictive control," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 10 549–10 556.
- [30] H.-J. Yoon, C. Tao, H. Kim, N. Hovakimyan, and P. Voulgaris, "Sampling complexity of path integral methods for trajectory optimization," in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 3482–3487.
- [31] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.
- [32] Y. Lee, J. Baek, Y. M. Kim, and F. C. Park, "Imat: The iterative medial axis transform," in *Computer Graphics Forum*, vol. 40, no. 6. Wiley Online Library, 2021, pp. 162–181.
- [33] W. Amanhoud, M. Khoramshahi, and A. Billard, "A dynamical system approach to motion and force generation in contact tasks," in *Robotics: Science and Systems (RSS)*, 2019.
- [34] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020. [Online]. Available: <https://doi.org/10.1007/s12532-020-00179-2>
- [35] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, "CasADi – A software framework for nonlinear optimization and optimal control," *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.
- [36] S. S. Mirrazavi Salehian, N. Figueroa, and A. Billard, "A unified framework for coordinated multi-arm motion planning," *The International Journal of Robotics Research*, vol. 37, no. 10, pp. 1205–1232, 2018.
- [37] M. Koptev, N. Figueroa, and A. Billard, "Real-time self-collision avoidance in joint space for humanoid robots," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1240–1247, 2021.
- [38] J. Carpentier, G. Saurel, G. Buondonno, J. Mirabel, F. Lamiraux, O. Stasse, and N. Mansard, "The pinocchio c++ library – a fast and flexible implementation of rigid body dynamics algorithms and their analytical derivatives," in *IEEE International Symposium on System Integrations (SII)*, 2019.
- [39] J. Pan, S. Chitta, and D. Manocha, "Fcl: A general purpose library for collision and proximity queries," in *2012 IEEE international conference on robotics and automation*. IEEE, 2012, pp. 3859–3866.
- [40] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.
- [41] M. Koptev, N. Figueroa, and A. Billard, "Reactive collision-free motion generation in joint space via dynamical systems and sampling-based mpc," *The International Journal of Robotics Research*, vol. 43, no. 13, pp. 2049–2069, 2024.