Complex In-Hand Manipulation via Compliance-Enabled Finger Gaiting and Multi-Modal Planning

Andrew S. Morgan[1*], Kaiyu Hang[2*], Bowen Wen[3], Kostas Bekris[3], and Aaron M. Dollar[1]
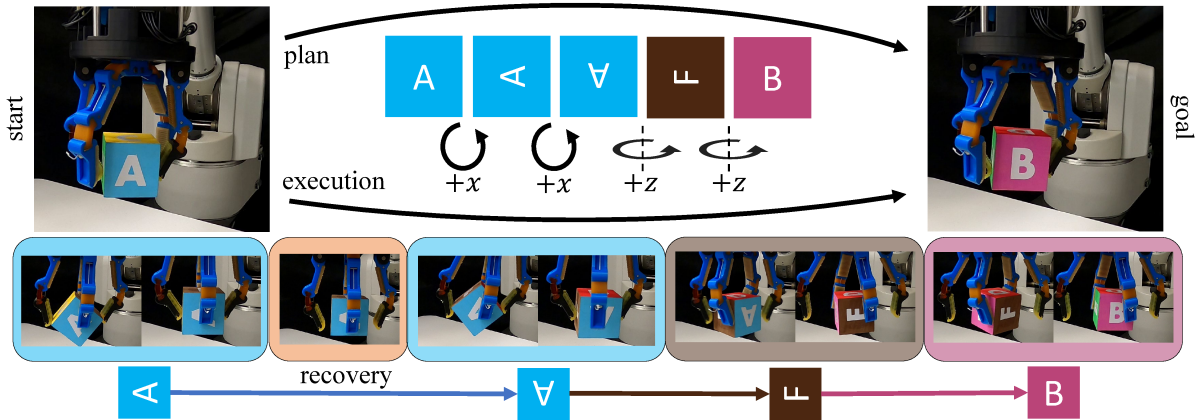
arXiv:2201.07928v1 [cs.RO] 20 Jan 2022

Fig. 1. We explore the development of a complete $SO(3)$ planner for within-hand manipulation using finger gaits, by controlling two orthogonal extrinsic rotation axes. Given a start configuration (cube face A), the proposed planner finds an action sequence along the two controlled dimensions so as to reach the desired goal configuration (cube face B). During manipulation, the pose of the object is tracked via a low-latency, 6D pose object tracker, providing feedback for online replanning and disturbance compensation via a recovery phase that uses translation control.

Abstract—Constraining contacts to remain fixed on an object during manipulation limits the potential workspace size, as motion is subject to the hand's kinematic topology. Finger gaiting is one way to alleviate such restraints. It allows contacts to be freely broken and remade so as to operate on different manipulation manifolds. This capability, however, has traditionally been difficult or impossible to practically realize. A finger gaiting system must simultaneously plan for and control forces on the object while maintaining stability during contact switching. This work alleviates the traditional requirement by taking advantage of system compliance, allowing the hand to more easily switch contacts while maintaining a stable grasp. Our method achieves complete $SO(3)$ finger gaiting control of grasped objects against gravity by developing a manipulation planner that operates via orthogonal safe modes of a compliant, underactuated hand absent of tactile sensors or joint encoders. During manipulation, a low-latency 6D pose object tracker provides feedback via vision, allowing the planner to update its plan online so as to adaptively recover from trajectory deviations. The efficacy of this method is showcased by manipulating both convex and non-convex objects on a real robot. Its robustness is evaluated via perturbation rejection and long trajectory goals. To the best of the authors' knowledge, this is the first work that has autonomously achieved full $SO(3)$ control of objects within-hand via finger gaiting and without a support surface, elucidating a valuable step towards realizing true robot in-hand manipulation capabilities.

[1]Andrew Morgan and Aaron Dollar are with the Department of Mechanical Engineering & Materials Science, Yale University, USA. (email: `Andrew.Morgan, Aaron.Dollar`}@yale.edu)

[2]Kaiyu Hang is with the Department of Computer Science, Rice University, USA. (email: `kaiyu.hang@rice.edu`)

[3]Bowen Wen and Kostas Bekris are with the Department of Computer Science, Rutgers University, USA. (email: {`bw344, kostas.bekris`}@cs.rutgers.edu)

Index Terms—Dexterous Manipulation, In-hand Manipulation, Manipulation Planning, Compliant Joints and Mechanisms

I. INTRODUCTION

WITHIN-HAND MANIPULATION can be characterized as the ability to reorient or reposition an object with respect to the hand frame [1]. The quest for this capability has been studied in the robot manipulation community for decades–from planning and control with rigid hands [2] to efforts with soft, compliant, or underactuated hands [3]. Notably, the majority of these works constrain contacts to remain fixed or rolling during manipulation. This constraint limits the object's workspace as the actuators can only operate on a single hand-object configuration manifold. Alternatively, finger gaiting, i.e., the process of repositioning contacts on the object during manipulation, can help alleviate such limitations and extend the object's available workspace.

Finger gaiting is an inherently difficult task for a robot. Given a robot hand, the individual serial link fingers must work in proper unison without collision; maintaining stability while making and breaking contact with the object [4]. The computational complexity of this problem has traditionally been very expensive–requiring the system to calculate and modulate forces and planned joint trajectories online during manipulation. We, conversely, are able to alleviate many of these complexities by leveraging compliance, i.e. safe modal transitions, in our system. The capabilities we present extend beyond what has been illustrated previously in the literature, some purely in simulation [5], [6] and some demonstrated on a real robot, but with support surfaces [7], [8], [9] .

In this letter, we build off the observation that by leveraging the passive adaptive properties of an underactuated hand, we can convert traditional position, force, and stability control problems to a unified motion-only control paradigm, based on the compliant properties of the hand. Concretely, finger gaiting requires contacts to be constantly in motion. The presence and

thus activation of certain contacts constrains what forces the hand can impart onto the object. This phenomena creates the abstraction of modes, which can be conceptualized as different families of motion manifolds that are subject to the system's current set of constraints–typically in the form of gaining or losing contacts. Multi-modal planning within and between these constraints thus becomes a major focus of this work.

We in this work can constrain our planning approach according to the nature of the task. Formally, any orientation in $SO(3)$ can be achieved via a three action trajectory comprised of two orthogonal rotations. From this formulation, we design two core modal actions for the hand and a multi-modal planner that runs online and is constantly updated via a vision-based, low-latency 6D pose object tracker [10], [11]. This method continually calculates a trajectory from start to goal and when undesired scenarios arise, such as contact slip, the planner updates and suggests new actions accordingly. We incorporate this approach into an open-source and underactuated hand with four fingers [12]. In the end, we showcase the efficacy of our system through various experiments: tracking the planned and executed trajectory of an object, evaluating the recovery potential given undesired perturbations, and finally, the ability to extend to novel object geometries.

The contributions in this letter are threefold. First, we develop a *complete* and fast planning solution for in-hand reorientation using two extrinsic rotation axes. Secondly, we describe the utility of compliance for switching between modes and how it "inflates" the contact switching region. And, finally, we present a simple yet effective robot system capable of complex finger gaiting capabilities, underscoring continued discussion in the community on the utility of compliance for in-hand manipulation tasks.

II. RELATED WORK

1) Modeling Manipulation: Modeling robot manipulation has historically been arduous as the dynamics associated with contact are difficult to predict in novel scenarios. From an in-hand manipulation perspective, various levels of modeling have been investigated – from contact models [13] and fingerpad curvature models [14], to hand kinematic models and whole hand-object system models [15]. While these approaches have elucidated many powerful hand-object relationships, inaccurate parameterizations can often lead to task failure [16]. To help alleviate such uncertainty, several end effectors leverage designs that are soft or underactuated [17]. This property provides passive reconfigurability to the mechanism, which can absorb much of the "slack" traditionally required to be fully accounted for in system modeling and can further reduce grasp planning times [18], [19]. In this work, we leverage such mechanisms to unify our planning approach–creating the notion of safe regions for contact switching.

2) Extending In-hand Manipulation Capabilities: Models for in-hand manipulation are typically constrained to fixed or rolling contact scenarios. While this advantageously simplifies assumptions for control, it also limits the object's available workspace to be dependent on the kinematic topology of the hand. Without relying on external contact, e.g, [20], or task-specific hands with roller-based fingers, e.g [21], two general approaches can help alleviate such constraint: sliding manipulation and finger gaiting. Leveraging the former is very difficult, as detecting and controlling its nonlinear conditions requires various levels of advanced sensing [22], [23]. The latter, alternatively, has been largely difficult due to computational considerations, but has been made more successful in recent years [24], [6]. The work in [25] used finger gaits and tactile sensors to maintain grasp stability. [7] used a 24-DOF hand with a motion capture system, in addition to "over 100 years" of simulated data to perform impressive and fast cube manipulation in the palm. [8] leveraged the capabilities of a soft hand with 16 degrees of actuation for similar types of manipulation. While as impressive as these aforementioned works are, we are interested in extending beyond these capabilities to work against gravity, i.e. maintaining object stability without a support surface, and while utilizing a simple hand with very little onboard sensing.

3) Multi-Modal Planning for Manipulation: Problems in robot manipulation are frequently multi-modal, i.e. the seemingly continuous problems have an underlying discrete structure guided by constraints [26]. These constraints are typically imposed by the nature of contact, and by discretizing planning in terms of these manifolds, the planner search space is confined according to physical constraints [27]. Numerous sampling-based multi-modal planners have been described in the literature, which are able to generalize well, particularly in high-DOF scenarios. Though, sampling-based methods without informed exploration are vastly inefficient and suffer from time-complexities associated with over-exploration. Recent work has attempted to address this issue by using informed "leads" to guide exploration [28]. In our work, we build off these observations and constrain our planner's search according to physical properties of the $SO(3)$ rotation group. That is, our developed planner constrains the number and nature of mode switching to reduce planning time for continual updates during online execution.

III. PRELIMINARIES

In this section, we introduce the preliminaries associated with multi-modal motion planning. We first discuss constrained manifolds and then develop notation and terminology for modal switching leveraging compliance.

A. Constrained Within-Mode Planning

Consider a robot with configuration space, $\mathcal{Q} \subset \mathbb{R}^N$, where N is the number of joints and where $\mathcal{Q}$ can completely define the state of the robot. Now, consider an object with configuration space, $\mathcal{O} \subset SE(3)$, which is located on a support surface, e.g. table top. For simplicity, let's disregard other potential collisions in the environment. In free space, the current robot configuration, $q \in \mathcal{Q}$, is unconstrained and thus able to freely move. Though, when q causes links of the robot to come in contact with the object in its current configuration, $o \in \mathcal{O}$, a constraint is imposed on the system. The robot cannot push the object through the support surface. If, for instance, the goal of the robot is to slide the object along its support surface, the robot's motion is thus constrained to a contact configuration manifold, $\mathcal{M}$ (Fig. 2(a)).
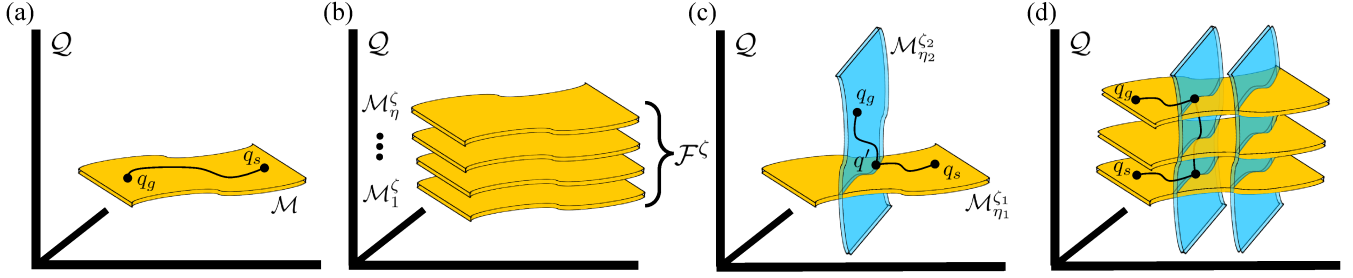
Fig. 2. Multi-modal planning problems can be conceptualized as operating in different configuration manifolds. (a) Given a single manifold $\mathcal{M}$, the planner must find a path along the constrained layer. (b) A single mode, or foliation, can have multiple manipulation manifolds depending on the start configuration of the hand-object system. (c) Switching modes is possible when the system finds a configuration, q', that lies on both manifolds. (d) Finding a path from start configuration, q_s, to goal configuration, q_g, can require multiple jumps between modes. Note that although represented in this figure as folds, we assume modes to be differentiable but not necessarily euclidean.

Assume there are several manifolds, indexed by η. The available motion manifold of the robot, $\mathcal{M}_\eta$, is subject to its associated constraint function $F^\eta(q, o) : \mathbb{R}^N \to \mathbb{R}^{k_\eta}$ where $(1 \leq k_\eta < N)$. Modes are thus defined as a family, or foliation, $\mathcal{F}^\zeta$, of constrained manifolds that share the same, or similar, constraint functions (Fig. 2(b)). Inside each $\mathcal{F}^\zeta$, we assume all $\mathcal{M}_\eta$ to be smooth and differentiable. Therefore, planning in the $\mathcal{M}_\eta \in \mathbb{R}^{(N-k_\eta)}$ manifold of $\mathbb{R}^N$ is subject to,

$$\mathcal{M}_\eta = \{q \in \mathcal{Q} | F^\eta(q, o) = \mathbf{0}\} \tag{1}$$

B. Safe Mode Transitions

Multi-modal planning requires a robot to transition between at least two constraint manifolds, which likely lie in different foliations, in order to reach the desired goal. Consider a robot starting in configuration q_s, located in manifold $\mathcal{M}_{\eta_1}^{\zeta_1}$. Let's now assume the goal configuration q_g is located in a different mode and thus along a different manifold, i.e. $\mathcal{M}_{\eta_2}^{\zeta_2}$ (Fig. 2(c)). In order for a robot to transition to q_g, a planner must find a candidate submanifold, $\mathcal{M}_{\eta_1 \cap \eta_2}$, that enables a transition. Inside of this submanifold lies candidate configuration, q', for transitioning. We can define $\mathcal{M}_{\eta_1 \cap \eta_2}$ according to,

$$\mathcal{M}_{\eta_1 \cap \eta_2} = \{q \in \mathcal{Q} | F^{\eta_1}(q, o) = \mathbf{0}, F^{\eta_2}(q, o) = \mathbf{0}\} \tag{2}$$

Note that it is possible for $\mathcal{M}_{\eta_1 \cap \eta_2}$ to be empty when there exists no q' to transition $\mathcal{M}_{\eta_1}^{\zeta_1}$ to $\mathcal{M}_{\eta_2}^{\zeta_2}$.

In many robot scenarios, if a transition is possible from $\mathcal{M}_{\eta_1}^{\zeta_1}$ to $\mathcal{M}_{\eta_2}^{\zeta_2}$, the transfer region typically has a volume near zero, i.e. $\Omega(\mathcal{M}_{\eta_1 \cap \eta_2}) \approx 0$ where Ω is the convex hull function. There are situations, however, where we can relax this constraint to form *safe mode transitions*. Formally, the manifold's constraint function, $F^\eta(q, o)$, is dependent on the robot's current configuration, q and object configuration, o. Though due to near-unmodelable DOFs in soft, compliant, or underactuated robots or objects, the controlled configuration of the robot may still hold for $\mathbb{R}^N \to \mathbb{R}^{k_\eta}$, but the true configuration of the system may be non-deterministic. Thus, there can exist a relaxation of this traditional modal switching constraint, which in turn can simplify and accelerate planning [18]. We define a modal transition threshold vector, ρ, that defines an inflation of the constraint manifold along different axes, which inherently increases the potential volume of $\mathcal{M}_{\eta_1 \cap \eta_2}$ to develop a *safe transfer region*,

$$\mathcal{M}_{\eta \cap \eta'}^* = \{q \in \mathcal{Q} | -\rho < F^\eta(q) < \rho, -\rho < F^{\eta'}(q) < \rho\} \tag{3}$$

The success of our finger gating platform in this work largely leverages this concept. Explicitly defining ρ, however, is difficult as it is system dependent, i.e. properties of the hand and the object determine how inflated the switching regions become. As a proof-of-concept analysis, we estimate ρ via experimentation in Sec. VII and will leave computational investigation for future work.

IV. ORIENTATION-BASED MOTION PLANNING

We formally define the problem of orientation-based motion planning and the minimum number of actions required. Then, we utilize this as a guarantee for the completeness our planner's modal search transitions.

A. Proper Rotations in $SO(3)$

Proper rotations are sets of angles that can fully define any rotation in the group, $SO(3)$. Let's assume we have an object with some rotation, R. It is possible to transition R to any rotation configuration via a combination of rotations along two orthogonal axes. Though there are six total combinations, we will instantiate our planner to focus on rotations along the $z-$axis (R_z) and $x-$axis (R_x) to serve as an example.

Theorem 1: *Let the rotational configuration of the object be $R \in SO(3)$. Then there exists at least one $(\phi, \theta, \psi) \in [0, 2\pi) \times [0, \pi] \times [0, 2\pi)$ such that $R = R_z(\phi)R_x(\theta)R_z(\psi)$ [29].*

B. Planning Orientation Transitions

Following this realization and the preliminaries in Sec. III, we develop a *complete* multi-modal planning algorithm for finger gaiting that consists of three rotations along two orthogonal axes. If, for instance, all three orthogonal axes were available, the planning problem would be mostly trivial–simply transition the object along each axis as much as necessary to reach the goal. For the duration of this paper, we will refer to rotations about axes as being extrinsic, i.e. according to the hand frame of robot, and will adopt the notation that $R_x(\cdot)$ and $R_z(\cdot)$ are rotations of the object about the $x-$ and $z-$axes of the hand frame, or roll and yaw, respectively.

Let's assume the capabilities of the robot enable the formulation of just two mode foliations, which are able to take an object in *any* configuration and rotate about the $x-$ and $z-$axes, respectively. Given an object in start configuration $o_s = \{T_s \in \mathbb{R}^3, R_s \in SO(3)\}$, the objective of this planner is to reach a desired goal pose, $o_g = \{T_g \in \mathbb{R}^3, R_g \in SO(3)\}$.
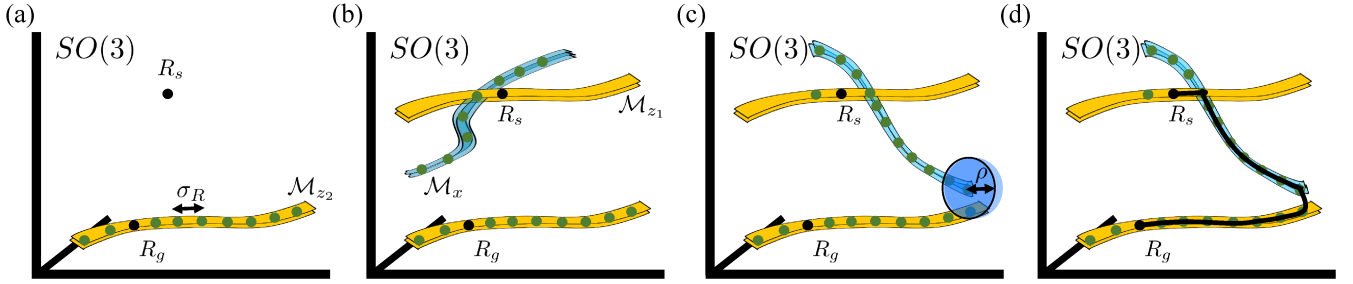
Fig. 3. (a) Our planning solution for $SO(3)$ begins by enumerating outward the goal orientation manifold by step size σ_R and creating a KD-Tree. (b) After creation, a forward search outward is performed from the start orientation, R_s, combining a $z-$axis rotation first, then an $x-$axis rotation. (c) This search continues until a candidate orientation is within some distance ρ from the expanded goal manifold. (d) Finally, the entire plan is enumerated from start to goal along the acquired trajectory with appropriate timestamps and step size σ_R.

Algorithm 1 $SO3Plan(\cdot)$

Input: R_s, R_g, ρ, σ_R ▷ object start orientation, object goal
     orientation, modal transition threshold, rotational step size
Output: $\mathcal{P}$ ▷ multi-modal plan
1: $\Sigma = [0, \sigma_R, -\sigma_R, 2\sigma_R, -2\sigma_R, \cdots, \pi, -\pi]$ ▷ candidate steps
2: $\mathcal{M}_{z_2} \leftarrow [R_z(\phi)R_g \ \textbf{for} \ \phi \ \textbf{in} \ \Sigma]$ ▷ expanded goal manifold
3: $KD \leftarrow KDTree.build(\mathcal{M}_{z_2})$ ▷ KD-Tree of goal manifold
4: **for** ψ **in** Σ **do**
5:      **for** θ **in** Σ **do**
6:          $R^* = R_x(\theta)R_z(\psi)R_s$ ▷ outward expanding search
7:          $\phi \leftarrow KD.query\_closest\_config(R^*)$
8:          **if** $dist_R(R_z(\phi)R_g, R^*) < \rho$ **then**   ▷ distance function
9:              **break loops**   ▷ bidirectional trajectory connects
10:          **end if**
11:      **end for**
12: **end for**
13: $\mathcal{P} \leftarrow$ **planner**.$enumerate(R_s, \sigma_R, \psi, \theta, \phi)$
       ▷ enumerate path outward according to connection (Fig. 3(d))
14: **return** $\mathcal{P}$

For now, we will disregard the translational positions, T_s and T_g, and focus on rotation.

As to accelerate computation, the proposed planning method is *bi-directional* in that it builds a search trajectory outward from both, the start and goal configurations. More formally, there exists rotations such that $R_g = R_{z_2}(\phi)R_x(\theta)R_{z_1}(\psi)R_s$ where $R_{z_1}(\psi)$ is the first rotation about the $z-$axis, $R_x(\theta)$ is the rotation about the $x-$axis, and $R_{z_2}(\phi)$ is the second rotation about the $z-$axis. Thus, the goal of this planner is to solve for ψ, θ, and ϕ and enumerate a trajectory accordingly.

Alg. 1 begins with the backward pass by populating a $z-$axis rotation outward from R_g with step size, σ_R, which is user-defined (Fig. 3(a)). For our implementation, σ_R can generally range anywhere from 0.5-5.0° and can be adaptive according to system architectures. Let's say that these dis-cretized rotations outward from R_g can sufficiently define the entire manifold, $\mathcal{M}_{z_2}$, i.e. an entire rotation of the $z-$axis about R_g. From these configurations in $\mathcal{M}_{z_2}$, we can build a KD-Tree, KD, which is a data structure that enables efficient querying of the rotations included in this manifold.

Now that the backwards direction is instantiated, the goal of the forward pass is to find rotations $R_x(\theta)R_{z_1}(\psi)R_s$ such that they can "connect" to the manifold $\mathcal{M}_{z_2}$ represented by KD within some distance threshold ρ from (3) (Fig. 3(c)). As aforementioned, ρ is the inflation factor for contact switching, and is generally system dependent. Here, we utilize a user-defined rotational distance function $dist_R(\cdot)$, where for our instantiation, we calculate the L^2-norm of the difference vector along the roll, pitch, and yaw dimensions of two rotations.

Though, we note that there are other metrics available, e.g. [30]. Tangibly, this forward pass sequentially searches through candidate values for ψ and θ until the trajectory reaches the goal manifold $\mathcal{M}_{z_2}$, which is found by continually querying KD. Once this connection is found (Fig. 3(d)), values for ψ, θ, and ϕ are recorded, and a trajectory for plan $\mathcal{P}$ is enumerated with timestamps along step size σ_R from start to goal.

V. OBJECT CONTROL

Finger-gaiting is a dynamic and uncertain process; making and breaking contacts can often lead to slip which could further result in object ejection. To mitigate such occurrences, our solution is to continually replan and execute modal actions. More specifically, we rely on an online servoing-based frame-work so as to reach the desired goal. When the object enters a region of potential instability, such as being distant from the workspace center, a recovery phase comprised of translational modal actions is enacted to relocate the object.

1) Translational Planning: Planning translations within-hand, presented in Alg. 2, performs a simple, greedy visual servoing control approach. Generally, the algorithm calculates the translational difference, γ, along each of the coordinate axes from start position, T_s, and goal position, T_g. The planner then chooses the direction of largest difference and creates a plan to take a single step in that direction [31].

Algorithm 2 $TranslationPlan(\cdot)$

Input: T_s, T_g, σ_T ▷ object position, object goal position, step size
Output: $\mathcal{P}$ ▷ plan
1: $\gamma.x \leftarrow T_g.x - T_s.x$ ▷ translational difference along $x-$axis
2: $\gamma.y \leftarrow T_g.y - T_s.y$ ▷ translational difference along $y-$axis
3: $\gamma.z \leftarrow T_g.z - T_s.z$ ▷ translational difference along $z-$axis
4: $\mathcal{P} \leftarrow \{\}$
5: $\pi \leftarrow max(|\gamma_x|, |\gamma_y|, |\gamma_z|)$ ▷ direction of maximum deviation
6: **if** π is $|\gamma_x|$ **then**
7:      $\mathcal{P} \leftarrow [T_x + \text{sgn}(\gamma_x)\sigma_T, T_y, T_z]$ ▷ desired move $\pm T_x$
8: **else if** π is $|\gamma_y|$ **then**
9:      $\mathcal{P} \leftarrow [T_x, T_y + \text{sgn}(\gamma_y)\sigma_T, T_z]$ ▷ desired move $\pm T_y$
10: **else if** π is $|\gamma_z|$ **then**
11:      $\mathcal{P} \leftarrow [T_x, T_y, T_z + \text{sgn}(\gamma_z)\sigma_T]$ ▷ desired move $\pm T_z$
12: **return** $\mathcal{P}$

---

2) Orientation and Translation Control: Our control ap-proach combines both, Alg. 1 and Alg. 2. This method requires a goal threshold, τ, a linear interpolant update parameter, λ, and *a priori* knowledge of the hand's workspace center, T_{center}, for object recovery.

In executing Alg. 3, orientation control is accomplished first. A plan $\mathcal{P}$ is found according to the object's current object

rotation, R. The robot then takes a *single* action along $\mathcal{P}$ via the **robot**.$mode\_action(\cdot)$ method. This system-specific function enacts a motor position step proportional to σ_R along the desired mode. From this action, an object pose displacement, δ, is calculated by taking the distance between the original orientation, R, and the new orientation, R', after the action. This value δ serves as a differential update term along the orientation transition, which allows us to adaptively re-estimate σ_R according to an interpolant update parameter λ. Since, as aforementioned, there is inherent uncertainty in the task of finger gaiting, this adaptive update to the transition step size σ_R serves to aid in our plan's accuracy for the true motion of the hand-object system. The algorithm continues while R is outside of the goal distance threshold τ_R. Notably, a recovery phase (lines 12-16) is developed inside of this loop and is enacted when the object position, T, is distance τ_T outside of T_{center}. At this point, the system takes greedy transition steps towards the center of the workspace for recovery.

Alg. 3 concludes by performing steps along $\mathcal{P}$ found from the $TranslationPlan(\cdot)$ method so as to reposition the object within-hand after object reorientation. We implement this method last in an attempt to decouple orientational and translational planning for the final pose.

Algorithm 3 $ObjectControl(\cdot)$

Input: o_g, ρ, σ, τ, λ, T_{center}
      ▷ object goal pose, modal transition threshold vector, pose transition step size, goal pose reached threshold, interpolant step size, center of object workspace
Output: o_f ▷ final object pose
1: $T_g, R_g \leftarrow o_g.T, o_g.R$ ▷ translation and rotation of goal pose
2: $o \leftarrow$ **tracker**.$perceive()$ ▷ object pose $\in SE(3)$
3: $T, R \leftarrow o.T, o.R$ ▷ translation and rotation of object pose

4: **while** $dist_R(R_g, R) > \tau_R$ **do**   ▷ orientation control
5:     $\mathcal{P} \leftarrow$ **planner**.$SO3Plan(R, R_g, \rho, \sigma_R)$ ▷ Alg. 1
6:     **robot**.$mode\_action(\mathcal{P}, \sigma_R)$ ▷ execute modal action
7:     $o' \leftarrow$ **tracker**.$perceive()$ ▷ new object pose
8:     $T', R' \leftarrow o'.T, o'.R$ ▷ new object translation and rotation
9:     $\delta \leftarrow dist_R(R', R)$ ▷ size of last action step
10:    $\sigma_R \leftarrow \sigma_R + \lambda(\sigma_R - \delta)$ ▷ update rotational transition step
11:    $T, R \leftarrow T', R'$
12:    **while** $dist_T(T, T_{center}) > \tau_T$ **do**   ▷ object recovery
13:       $\mathcal{P} \leftarrow$ **planner**.$TranslationPlan(T, T_{center}, \sigma_T)$
14:       **robot**.$mode\_action(\mathcal{P}, \sigma_T)$ ▷ execute modal action
15:       $o \leftarrow$ **tracker**.$perceive()$ ▷ object pose
16:       $T, R \leftarrow o.T, o.R$ ▷ object translation and rotation
17:    **end while**
18: **end while**

19: **while** $dist_T(T_g, T) > \tau_T$ **do**   ▷ translation planning
20:     $\mathcal{P} \leftarrow$ **planner**.$TranslationPlan(T, T_g, \sigma_T)$ ▷ Alg. 2
21:     **robot**.$mode\_action(\mathcal{P}, \sigma_T)$ ▷ execute modal action
22:     $o \leftarrow$ **tracker**.$perceive()$ ▷ object pose
23:     $T, R \leftarrow o.T, o.R$ ▷ object translation and rotation
24: $o_f \leftarrow$ **tracker**.$perceive()$ ▷ final object pose
25: **return** o_f

---

3) Generalization and Practical Algorithm Modifications: The authors present this solution as a generalized approach to planning for in-hand manipulation. Surely, the design and realization of the **robot**.$mode\_action(\cdot)$ method begs questions for system-specific scenarios, but can be found in a variety of ways: analytical modeling, dynamics learning, reinforcement learning, etc. While we do not focus on describing the un-
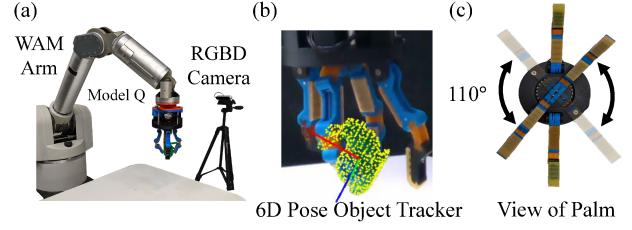

Fig. 4. (a) Our system is comprised of a RGBD camera fixated to the robot's environment. (b) During manipulation, the pose of the object, e.g. bunny, is tracked via a 6D pose object tracker. (c) A bottom view of the Yale Model Q illustrates the $110°$ abduction capabilities of the differentially coupled fingers.

derlying components of this function in this work, we largely leverage an energy-modeling technique outlined in [32] for determining predicted object transitions.

Moreover, the proposed planner can be further accelerated by subtle variations to the aforementioned algorithms. First, KD only needs to be computed once since the goal pose is static. Also, initialized step sizes for σ can deviate between modes according to system-specific properties, e.g. when gravity affects the rotation for one mode more than another.

VI. SYSTEM SETUP

1) Robot Setup: We utilize the open-source Model Q, an underactuated hand from the Yale OpenHand Project. The hand is equipped with four two-link fingers and four total actuators. A single motor controls the actuation of an opposing set of fingers with joints comprised of flexures, guided by a differential. This enables passive reconfigurability between these two fingers and into/out-of the plane of grasping. These two coupled fingers are connected to an actuated rotary joint located within the palm, allowing $110°$ rotation about the $z-$axis of the hand (Fig. 4). The final two motors serve to actuate the remaining two fingers individually via tendons. Being underactuated, the hand's joint configuration cannot be determined directly, as it is not equipped with joint encoders or tactile sensors. See [12] for additional design information.

The hand is mounted on a 7-DOF Barrett WAM arm. External to the robot, an RGBD Intel Realsense D415 camera is calibrated to the robot's environment. Its imagery is processed through the low-latency (60Hz) 6D pose object tracker (Fig. 4). The tracker is robust to finger occlusion, has approximately $\pm 1.5mm/\ 2°$ of noise along any axis, and is trained solely with synthetic object data. See [10] for additional information.

2) Mode Design: As described in Sec. IV-B, a series of rotations about two orthogonal axes allow an object to reach any orientation in $SO(3)$. Given the hand topology of the Model Q, an extrinsic $z-$axis rotation is easily possible via the palm's rotary joint. Moreover, an extrinsic $x-$axis rotation is also possible via coordinated finger motions; grasping with the differentially actuated opposing pair and pushing the object from either individually driven fingers as validated by [32].

In addition to the two rotational motions, the hand topology furthermore allows for only two translational modes. The first can translate the object along the $z-$axis by squeezing the object towards the palm and regrasping, or by leveraging small amounts of coordinated slip, and simple $y-$axis translation is possible via a method described in [31] (Fig. 5). Note that translation about the $x-$axis is not possible due to the
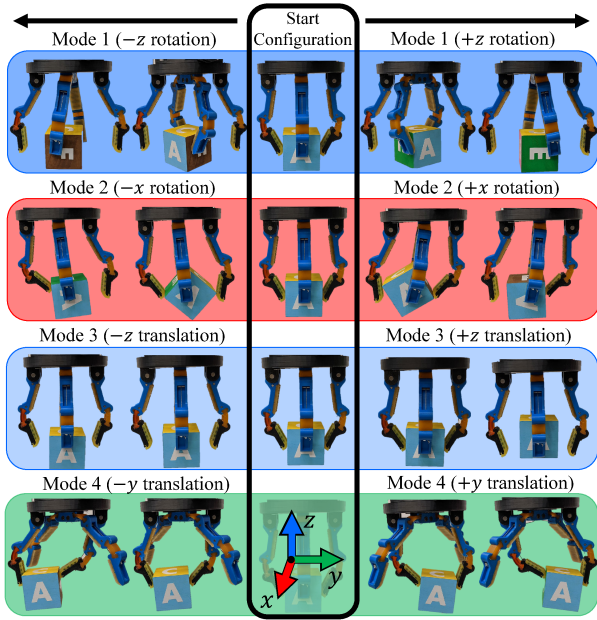
Fig. 5. The Yale OpenHand Model Q is capable of operating along four modes for fingertip-based manipulation. Modes are shown with transitions from the start configuration (center).

Fig. 6. (a) We experimentally validate our method with 5 different objects, namely a cube, sphere, toy truck, Stanford Bunny, and toy duck. (b) Tessellated faces of the cube show poses of the affixed letters.

actuation coupling of the rotary set of fingers. By combining these two translational modes, we are able to provide a recovery phase for the object (Fig. 1, 8), repositioning the object towards the middle of the workspace so that manipulation can safely continue. Note that $x-$axis translation is thus omitted.

VII. EXPERIMENTS

1) Safe Modes Characterization: The passive adaptive nature of compliant hands is particularly beneficial for the task of finger gaiting. Simply, small errors associated with state estimation, modeling, and control can likely be dismissed by the system "absorbing" the slack. It moreover determines the "inflation" of switching regions, denoted by ρ. To help quantify and validate this adaptive nature, we evaluate how a *regrasp* action, i.e. transferring the grasp between sets of opposing fingers, affects the resultant object configuration.

Evaluated with the cube (Fig. 6), we perturb the object along each axis. Subsequently, the hand transfers the grasp to the opposite set of opposing fingers. In doing so, we record the amount of object reconfiguration along each axis via the object tracker. Notably, with the roll orientation of the object, we see a small amount of reconfiguration, but in the yaw orientation, we note the most reconfiguration (Fig. 7), due to the geometric properties of the cube. More specifically, from the start grasp (blue circle), to the final regrasped configuration (red square), the object tends to fall towards the minimum energy configuration of the system (red region) [32], [24]. We note that although there is a large variability in the object's
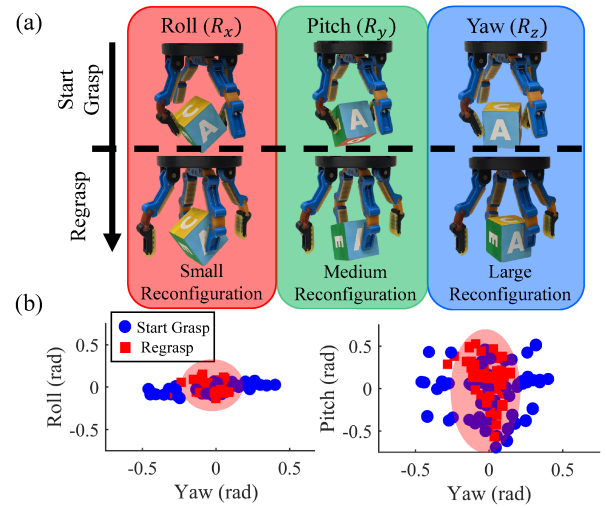

Fig. 7. The safety of mode transitions are attributed to the reconfigurability of the underactuated mechanism. (a) Starting rotations around different axes of the object elicit different amounts of reconfiguration upon regrasping. (b) This is particularly apparent in the yaw ($z-$axis) direction of the object, where the object follows the minimum energy configuration of the mechanism, and hence allowing us to estimate ρ from Eq. (3).

start orientation (± 0.5 radians), the hand is able to regrasp the object successfully in all cases. This underscores our *safe modes* discussion, and generally allows us to estimate how large ρ can be. For safety, we define $\rho = 0.2$ radians.

2) Single Trajectory Execution: We evaluate the repeatability of executing a single trajectory of the cube. Here, the goal of the task is to rotate the cube from face A to face B, and when completed, translate the object 1.8cm along the negative $y-$axis. The resultant executions, depicted in Fig. 8(a), shows the average trajectory of the object over 8 different trials. During this task, we note how closely trajectories follow along the roll and yaw dimensions, as these are controlled via our safe modes. Object recovery occurs at similar times in the trials, where the system transitions the object along the positive $z-$axis before completing the $x-$axis rotations (Fig. 8(b)). At the end of the orientation control sequence, the object is appropriately translated along the $y-$axis to reach its desired goal pose. The average trajectory data, depicted in solid black, does not extend past the 55 second mark due to one trial finishing earlier than others. The authors have, for clarity, extended the trajectory in the $y-$axis translation using a dotted line to illustrate the average final pose.

3) Continuous Goal Trajectories and Robustness: We showcase the robustness of our method by evaluating both, an extended control trajectory and by disturbing the object pose during execution. As depicted in Fig. 9(a), we execute a trajectory starting on cube face F, transitioning through faces A,B,C,D,E and finally reaching the face F once again. In all cases, the object was able to reach the goal face within 8°.

Moreover, we illustrate our ability to replan control trajectories when unmodeled perturbations occur (Fig. 9(b)). At different points during manipulation, we randomly perturb the object pose 13 total times along the least constrained dimensions, the $x-$axis rotation and the $z-$axis translation. Recovering and adapting to these occurrences over the course of a 205 second trajectory execution, the system completes the object reorientation task within 6° of the goal face.
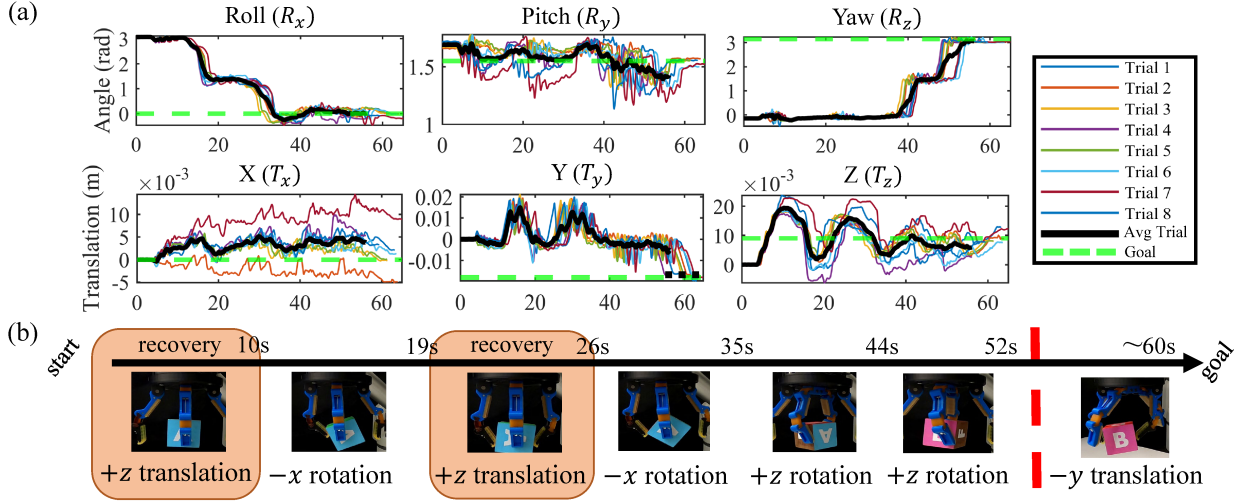
Fig. 8. We execute a single planned trajectory 8 times and record its repeatability. (a) Controlled dimensions of the object trajectory, such as roll and yaw, follow closely in all trials, where as uncontrolled dimensions, such as x−axis translation and pitch, are allowed to drift. (b) Modes are enacted at different times in the manipulation, including recovery phases when the object is not in the center of the graspable workspace.
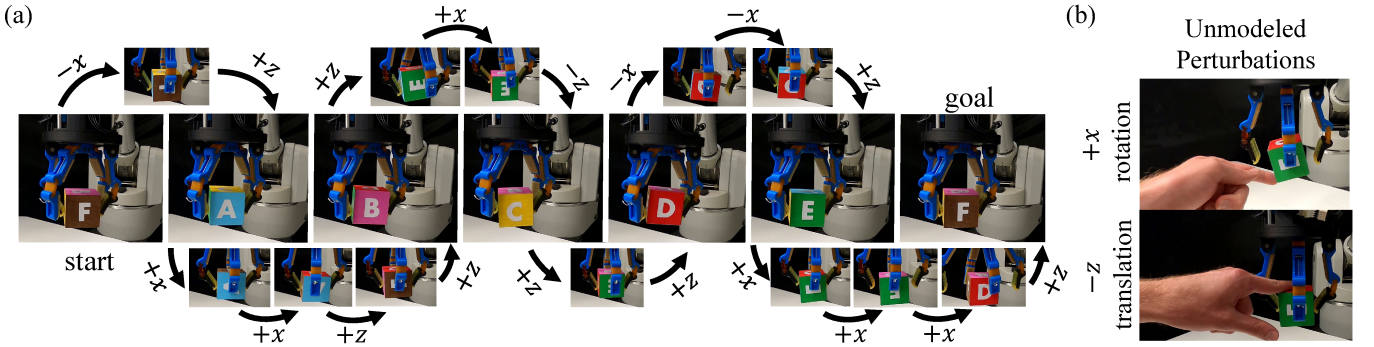

Fig. 9. We test system robustness by (a) following through an extended trajectory of cube faces F, A, B, C, D, E, and F once again, and (b) deliberately applying different perturbations to the object along controlled dimensions so as to require online replanning and recovery.
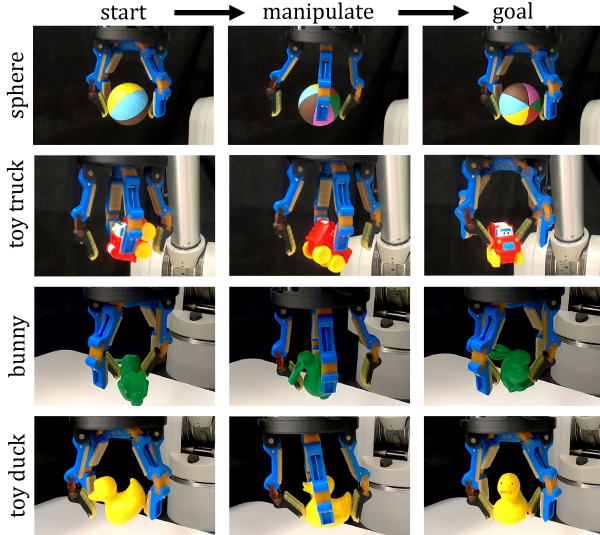

Fig. 10. Our manipulation planner is able to extend to objects of convex and non-convex geometries, and reach any orientation in $SO(3)$.

Obj.	err (deg)	Plan time (s)	Total time (s)	Success
sphere	4.6	13.2	153.1	5/5
truck	4.2	10.8	112.3	5/5
bunny	9.5	14.0	128.0	2/5
duck	8.9	7.1	73.2	3/5

Fig. 11. Experimental evaluation of four different objects with error in final orientation, planning time, total time, and number of execution successes.

4) Generalization to Object Geometry: Finally, we test the transferability of our method by varying object geometries (Fig. 6). In doing so, recovery properties, such as T_{center} from Alg. 3, were recalculated, and the 6D pose object tracker was retrained with new synthetic object data; other aforementioned parameters, including ρ, remained the same. Depicted in Fig.

10, we evaluate simple, purely rotational trajectories about each of the objects. Notably, the sphere easily followed each of its guided reference trajectories, and was able to reach goal orientations within $\pm 5°$ along any axis. The toy truck, further illustrated our capabilities, transitioning to its goal configuration in approximately 110 seconds with 4 recovery phases. And finally, the Stanford Bunny and the toy duck were the most difficult to manipulate due to their non-convex properties. With the bunny, fingers would often get stuck behind its ears while making and breaking contact. Consequently, this occurrence caused the hand-object system to get stuck in specific object orientations and required restart. Moreover, during duck manipulation, the size of the duck's head affected the object's center of mass. This made some actions, such as x−axis rotation, difficult as the motion was then more dynamic in nature. Only a simple, two-rotation trajectory could be completed with the duck, resulting in a shorter execution time. The precision of these two manipulations was decreased to $\pm 10°$ along any rotation axis and were consequently not

quite as robust as the other objects. We provide benchmarking baselines for our experimentation in Fig. 11 according to guidelines outlined [33]. Please refer to the supplementary video for experimental evaluations.

VIII. DISCUSSIONS AND FUTURE WORK

In this work, we developed a robust and *complete* solution to $SO(3)$ finger gait planning. While instantiated on an underactuated hand in this work, the methods described can generalize to other hand-object systems. We showcased our method with numerous tasks–highlighting the safety of our modes, the repeatability of our trajectories, the robustness of our planning and control approach, and our ability to generalize to new object scenarios. This work builds on a promising approach of vision-based control for compliant within-hand manipulation. The tasks completed in this letter extend beyond what has been possible in previous works.

Although we showcase novel capabilities, there are some limitations. First, hand design plays a crucial role in our manipulation capabilities. We conceptualize an altered hand that would enable rotation about the $y-$axis, thus shortening trajectories. While our method may have relied less on data and/or advanced sensing, we note that our manipulation actions were significantly slower than some other related works [7], [8]. Moreover, we discuss the "inflation" parameter ρ quite extensively, which begs for further theoretical investigation. To this end, the development of geometry-focused modal actions and transitions is an interesting avenue. Lastly, the current tracking schema requires an object CAD model for training. We are interested in combining object-agnostic tracking methods [34] together with the current object-agnostic controller for instant application to novel objects.

REFERENCES

[1] A. Bicchi, "Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity," *IEEE Transactions on robotics and automation*, vol. 16, no. 6, pp. 652–662, 2000.

[2] J. Kerr and B. Roth, "Analysis of multifingered hands," *The International Journal of Robotics Research*, vol. 4, no. 4, pp. 3–17, 1986.

[3] A. Sintov, A. S. Morgan, A. Kimmel, A. M. Dollar, K. E. Bekris, and A. Boularias, "Learning a state transition model of an underactuated adaptive hand," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1287–1294, 2019.

[4] L. Han and J. C. Trinkle, "Dextrous manipulation by rolling and finger gaiting," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, vol. 1. IEEE, 1998, pp. 730–735.

[5] B. Sundaralingam and T. Hermans, "Geometric in-hand regrasp planning: Alternating optimization of finger gaits and in-grasp manipulation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 231–238.

[6] G. Khandate, M. Haas-Heger, and M. Ciocarlie, "On the feasibility of learning finger-gaiting in-hand manipulation with intrinsic sensing," 2021.

[7] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray *et al.*, "Learning dexterous in-hand manipulation," *The International Journal of Robotics Research*, vol. 39, no. 1, pp. 3–20, 2020.

[8] A. Bhatt, A. Sieler, S. Puhlmann, and O. Brock, "Surprisingly robust in-hand manipulation: An empirical study-supplementary material," 2021.

[9] S. Abondance, C. B. Teeple, and R. J. Wood, "A dexterous soft robotic hand for delicate in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5502–5509, 2020.

[10] B. Wen, C. Mitash, B. Ren, and K. E. Bekris, "se (3)-tracknet: Data-driven 6d pose tracking by calibrating image residuals in synthetic domains," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10367–10373.

[11] B. Wen, C. Mitash, S. Soorian, A. Kimmel, A. Sintov, and K. E. Bekris, "Robust, occlusion-aware pose estimation for objects grasped by adaptive hands," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6210–6217.

[12] R. R. Ma and A. M. Dollar, "An underactuated hand for efficient finger-gaiting-based dexterous manipulation," in *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. IEEE, 2014, pp. 2214–2219.

[13] J. C. Trinkle, J.-S. Pang, S. Sudarsky, and G. Lo, "On dynamic multi-rigid-body contact problems with coulomb friction," *Journal of Applied Mathematics and Mechanics*, vol. 77, no. 4, pp. 267–279, 1997.

[14] D. J. Montana, "Contact Stability for Two-Fingered Grasps," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 4, pp. 421–430, 1992.

[15] R. M. Murray, S. S. Sastry, and L. Zexiang, *A Mathematical Introduction to Robotic Manipulation*. USA: CRC Press, Inc., 1994.

[16] K. Hang, W. G. Bircher, A. S. Morgan, and A. M. Dollar, "Manipulation for self-identification, and self-identification for better manipulation," *Science Robotics*, vol. 6, no. 54, 2021.

[17] A. M. Dollar and R. D. Howe, "The Highly Adaptive SDM Hand: Design and Performance Evaluation," *The International Journal of Robotics Research*, vol. 29, no. 5, pp. 585–597, apr 2010.

[18] K. Hang, A. S. Morgan, and A. M. Dollar, "Pre-grasp sliding manipulation of thin objects using soft, compliant, or underactuated hands," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 662–669, 2019.

[19] A. Morgan, B. Wen, J. Liang, A. Boularias, A. Dollar, and K. Bekris, "Vision-driven compliant manipulation for reliable; high-precision assembly tasks," *Robotics: Science and Systems XVII*, Jul 2021.

[20] R. H. N. Chavan-Dafle and A. Rodriguez, "Planar in-hand manipulation via motion cones," *IJRR*, 2019.

[21] S. Yuan, L. Shao, C. L. Yako, A. Gruebele, and J. K. Salisbury, "Design and control of roller grasper v2 for in-hand manipulation," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 9151–9158.

[22] D. L. Brock, "Enhancing the dexterity of a robot hand using controlled slip," in *Proceedings. 1988 IEEE International Conference on Robotics and Automation*. IEEE, 1988, pp. 249–251.

[23] Z. Su, K. Hausman, Y. Chebotar, A. Molchanov, G. E. Loeb, G. S. Sukhatme, and S. Schaal, "Force estimation and slip detection/classification for grip control using a biomimetic tactile sensor," in *2015 IEEE-RAS 15th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2015, pp. 297–303.

[24] W. G. Bircher, A. S. Morgan, and A. M. Dollar, "Complex manipulation with a simple robotic hand through contact breaking and caging," *Science Robotics*, vol. 6, no. 54, 2021. [Online]. Available: https://robotics.sciencemag.org/content/6/54/eabd2666

[25] K. Hang, M. Li, J. A. Stork, Y. Bekiroglu, F. T. Pokorny, A. Billard, and D. Kragic, "Hierarchical fingertip space: A unified framework for grasp planning and in-hand grasp adaptation," *IEEE Transactions on robotics*, vol. 32, no. 4, pp. 960–972, 2016.

[26] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *The International Journal of Robotics Research*, vol. 29, no. 7, pp. 897–915, 2010.

[27] M. Zucker, N. Ratliff, A. D. Dragan, M. Pivtoraiko, M. Klingensmith, C. M. Dellin, J. A. Bagnell, and S. S. Srinivasa, "Chomp: Covariant hamiltonian optimization for motion planning," *The International Journal of Robotics Research*, vol. 32, no. 9-10, pp. 1164–1193, 2013.

[28] Z. Kingston, A. M. Wells, M. Moll, and L. E. Kavraki, "Informing multi-modal planning with synergistic discrete leads," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 3199–3205.

[29] J. J. Craig, *Introduction to robotics: mechanics and control, 3/E*. Pearson Education India, 2009.

[30] D. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, pp. 155–164, 10 2009.

[31] B. Calli and A. M. Dollar, "Vision-based model predictive control for within-hand precision manipulation with underactuated grippers," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2839–2845.

[32] A. S. Morgan, K. Hang, and A. M. Dollar, "Object-agnostic dexterous manipulation of partially constrained trajectories," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5494–5501, 2020.

[33] S. Cruciani, B. Sundaralingam, K. Hang, V. Kumar, T. Hermans, and D. Kragic, "Benchmarking in-hand manipulation," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 588–595, 2020.

[34] B. Wen and K. Bekris, "Bundletrack: 6d pose tracking for novel objects without instance or category-level 3d models," *IROS*, 2021.