2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Torque-Limited Manipulation Planning through Contact by Interleaving Graph Search and Trajectory Optimization

Ramkumar Natarajan[1], Garrison L.H. Johnston[2], Nabil Simaan[2], Maxim Likhachev[1] and Howie Choset[1]

Abstract— **Robots often have to perform manipulation tasks in close proximity to people (Fig 1). As such, it is desirable to use a robot arm that has limited joint torques so as to not injure the nearby person. Unfortunately, these limited torques then limit the payload capability of the arm. By using contact with the environment, robots can expand their reachable workspace that, otherwise, would be inaccessible due to exceeding actuator torque limits. We adapt our recently developed INSAT algorithm [1] to tackle the problem of torque-limited whole arm manipulation planning through contact. INSAT requires no prior over contact mode sequence and no initial template or seed for trajectory optimization. INSAT achieves this by interleaving graph search to explore the manipulator joint configuration space with incremental trajectory optimizations seeded by neighborhood solutions to find a dynamically feasible trajectory through contact. We demonstrate our results on a variety of manipulators and scenarios in simulation. We also experimentally show our planner exploiting robot-environment contact for the pick and place of a payload using a Kinova Gen3 robot. In comparison to the same trajectory running in free space, we experimentally show that the utilization of bracing contacts reduces the overall torque required to execute the trajectory.**

I. Introduction

Collaborative robots can reduce the physiological burden of physically demanding tasks for human operators working in confined spaces. These robots can assist humans by manipulating heavy payloads deep inside a confined space. For such tasks, these long-reach robots need large torque actuators and massive links to support its own weight along with the payload to operate in configurations near its maximum reach. However, such operational requirements compromise the safety of collaboration with the human worker at close proximity. As a result, we are faced with a manipulation planning problem where the planner should minimize the manipulator joint torques and accelerations while respecting task manipulation requirements and avoiding obstacles.

To overcome the conflicting requirements of safe collaboration and operation in deep confined spaces, it has been shown that the robot can brace against the environment to reduce the overall effort required to manipulate heavy objects [2]. The physical constraints imposed by the environment can be transformed into opportunities that can be exploited to enable efficient manipulation that expends low energy, increases accuracy [3], [4], and reduces compliance [5].

In this work, we present a motion planning algorithm for manipulation that automatically discovers and exploits bracing locations along the entire trajectory to achieve a desired task such as transporting an overweight payload. Consequently, our torque-limited manipulation planning algorithm can opportunistically make/break/sustain contact with the environment to reach deep inside a confined space with insufficient actuator torques or carry a heavy payload beyond the manipulator's capability.

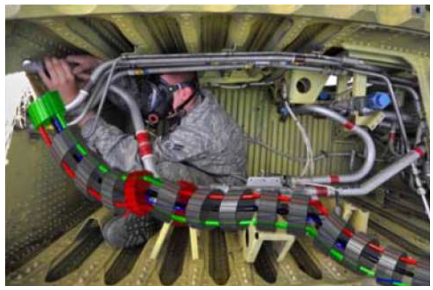

Fig. 1: An example of a hyperredundant robot manipulator lifting a heavy tool in a confined space by leveraging contact with the environment to assist a human worker.

The summary of our contributions is as follows:

- A novel adaptation of INSAT: INterleaved Search And Trajectory optimization [1] for the application of torque-limited manipulation planning through contact. By *interleaving* discrete graph-search with continuous trajectory optimization, our algorithm is able to plan through contact over long horizons for high-dimensional complex manipulation problems in confined non-convex environments.
- A dynamically feasible trajectory through contact can be non-smooth with impacts and discontinuities. We introduce a new virtual contact frictional force model to enable planning for complex, contact-rich motions without relying on a pre-specified contact schedule using a gradient-based optimizer.
- To the best of our knowledge, manipulation planning that actively reasons about effort reduction by utilizing additional support from contact has not been proposed and demonstrated on a real robot arm until now, which forms the most important contribution of this work.

The key idea behind our framework is (a) to identify a low-dimensional manifold, (b) perform a search over a grid-based graph that discretizes this manifold, and (c) while searching the graph, utilize contact-implicit trajectory optimization to compute the cost of partial solutions found by the search. As a result, the search over the lower-dimensional graph decides what trajectory optimizations to run and with what seeds, while the cost of the solution from the trajectory optimization drives the search in the lower-dimensional graph until a dynamically feasible trajectory from start to goal is found.

This paper is structured as follows: we discuss prior work

[1] The authors are with The Robotics Institute at Carnegie Mellon University, Pittsburgh PA 15213. email: {rnataraj, maxim, choset}@cs.cmu.edu

[2] The authors are with the department of Mechanical Engineering at Vanderbilt University. email: {garrison.l.johnston, nabil.simaan}@vanderbilt.edu

This work was supported by NSF awards #1734461 and #1734460, ARL grant W911NF-18-2-0218 and by Vanderbilt and Carnegie Mellon internal university funds.

2023 IEEE International Conference on Robotics and Automation (ICRA) | 979-8-3503-2365-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICRA48891.2023.10161297

in Sec. II, formalize the problem statement in III and introduce the tunable virtual contact models for contact-implicit trajectory optimization in IV. We then describe our proposed method in Sec. V. Finally, we show the experimental results in Sec. VI, and conclude in Sec. VII.

II. Prior Work

Although the idea of bracing against the environment for manipulation was proposed as early as the 1980s [6], not much attention has been paid since then. Sensing and control for bracing with probabilistic contact estimation [7] and multiple simultaneous contacts with the environment [8] were introduced in the 2000s. The methods proposed in [9] and [2] from the late 2010s were the first to consider manipulation planning through bracing against the environment. However, [2] is primarily a control algorithm to drive the robot to the right contact point and posture and not a planner that reasons about bracing to achieve a desired long-term task. Whereas the idea presented in [9] is limited in that it (i) ignores the contact dynamics (ii) only considers static environmental bracing and (iii) is demonstrated only on a simple planar elastomer manipulator with a single 2D obstacle.

In contrast, our planner generates trajectories with dynamic bracing (contact sliding); is evaluated in a much more challenging environment; and is demonstrated on a real robot. Planning to brace and navigating through the environment by bracing can be construed as constrained manipulation planning over a sub-manifold which has been addressed by [10] and its variants. But these are quasi-static methods that do not factor in the dynamics of contact and manipulator system essential to understand and leverage the effects of bracing.

Planning through contact with unspecified mode sequence is an active challenge in robot locomotion and manipulation [11]. The aim of the general formulation, called Contact Implicit Trajectory Optimization (CITO), is to jointly find trajectories for state, control input, and contact forces. Most of the successful previous works propose different combinations of trajectory optimization-based approaches [12], including direct shooting [13] and direct transcription [14], [15]. For incorporating contact, they use either complementarity conditions with implicit time-stepping [14], [15] or soft constraints implemented as a penalty term [16], [17] in the cost function [18], [19]. However, standalone optimization-based approaches are brittle when it comes to global reasoning over long horizon and depends heavily on the quality of initial guesses.

On the other hand, the contact mode sequence is inherently discrete and the optimizer faces a fundamentally discrete choice at each time, which is difficult to optimize whether modeled using continuous constraints or integer variables. To that end, recent approaches use graph search-based methods [20], [21], [22] or rapidly-exploring random trees [23] to plan contact switches and generate a seed for the subsequent trajectory optimization. However, these local methods are greedy and do not offer a fall-back in case the trajectory optimization does not succeed using the discrete contact sequence. In contrast, INSAT offers a principled way to globally reason over the discrete and continuous parts of the problem.

III. Problem Statement

In this work, we denote the robot manipulator as $\mathcal{R}$, and $\mathcal{X}^{\mathcal{R}} \subseteq \mathbb{R}^N$ as the configuration space (C-space) for a N degree-of-freedom (DoF) manipulator. Let $\mathcal{X}^{\text{obs}} \subset \mathcal{X}^{\mathcal{R}}$ be the C-space obstacle, $\mathcal{X}^{\text{free}} = \mathcal{X}^{\mathcal{R}} \setminus \mathcal{X}^{\text{obs}}$ be the free space and $\mathcal{X}^S \subset \mathcal{X}^{\text{obs}}$ denote the surface of the obstacle with which the robot can make and break contact. The planning state is comprised of joint angles and joint velocities $\mathbf{x} = [\mathbf{q}, \dot{\mathbf{q}}] \in \mathcal{X} \subseteq \mathbb{R}^{2N}$. The manipulator is controlled by bounded joint torque inputs $\mathbf{u} \in \mathbb{R}^N$. Given (a) a start state $\mathbf{x}^S$, (b) a goal state $\mathbf{x}^G$, (c) the planning space $\mathcal{X}$ with the obstacles $\mathcal{X}^{\text{obs}}$ and the obstacle contact surface $\mathcal{X}^S$, the task is to find a control trajectory $\mathbf{u}(t);\ t \in [0, T]$ according to Eq. 1.

For torque-limited planning, the manipulator's maximum velocity and torque constraints must be satisfied while planning. The energy-optimal motion planning for torque limited manipulation can be cast as the following optimization:

$$\begin{aligned} \text{find} \quad & \mathbf{u}(t) \\ \text{s.t.} \quad & \mathbf{x}(t) = \mathbf{f}(\mathbf{x}^S, \mathbf{u}(t)), \\ & \mathbf{x}(T) = \mathbf{x}^G \\ & \mathbf{x}(t) \in (\mathcal{X} \setminus \mathcal{X}^{\text{obs}}) \cup \mathcal{X}^S \\ & |\dot{\mathbf{x}}(t)| \leq \dot{\mathbf{x}}_{\text{lim}}, |\ddot{\mathbf{x}}(t)| \leq \ddot{\mathbf{x}}_{\text{lim}}, |\mathbf{u}(t)| \leq \mathbf{u}_{\text{lim}} \end{aligned} \tag{1}$$

where $\mathbf{f}$ denotes the manipulator dynamics with contact that captures the interaction of $\mathcal{R}$ with the environment (Eq. 2). Note that $\mathbf{x}(t)$ can lie on $\mathcal{X}^S$ and hence encodes the sequence of making and breaking contact with environment.

IV. Manipulator and Contact Model Dynamics

We model the dynamics of $\mathcal{R}$ and its interaction with the environment as a rigid-multi-body system using Euler-Lagrange mechanics with generalized coordinates $\mathbf{q}$ as:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau} + \mathbf{J}_{\boldsymbol{\Omega}}(\mathbf{q})^{\mathsf{T}}\boldsymbol{\Omega} \tag{2}$$

where $\mathbf{M}$, $\mathbf{C}$, $\mathbf{G}$ are mass, Coriolis and gravity matrices, $\boldsymbol{\tau}$ is the generalized input, $\mathbf{J}_{\boldsymbol{\Omega}}(\mathbf{q})$ is the contact Jacobian that maps $\dot{\mathbf{q}}$ to the Cartesian velocities at the external contact point, and $\boldsymbol{\Omega}$ is the contact forces.

In this work, we use MuJoCo [24] to simulate the manipulator dynamics with contact at high-fidelity. Contact introduces impacts and discontinuities in the system dynamics as the contact forces (*i.e.* $\boldsymbol{\Omega}$ from MuJoCo) vanish completely when not in contact and explode at the instant of making contact. A dynamically feasible control trajectory for our application might be non-smooth as the robot has to make/break/sustain contact with the environment. To optimize for such a trajectory using a gradient-based solver, we introduce two tunable smooth contact models. A smooth contact model is differentiable even at the collision event of the contact and enables faster trajectory optimization convergence. These models provide virtual forces that can be exploited in trajectory optimization to overcome the vanishing/exploding gradients of contact dynamics and enable automatic discovery of contact locations and smooth breaking of static friction. The vector of generalized joint inputs $\boldsymbol{\tau}$ can be decomposed as follows:

$$\boldsymbol{\tau} = \mathbf{u} - \mathbf{J}_{\boldsymbol{\Gamma}}(\mathbf{q})^{\mathsf{T}}\boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{q}}) \tag{3}$$

where $\mathbf{u}$ is the joint torque input, $\boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{q}}) \in \mathbb{R}^{N_\Gamma}$ and $\mathbf{J}_{\boldsymbol{\Gamma}}(\mathbf{q})$ are respectively the generalized virtual contact forces from the tunable smooth contact models (Eq. 4, 5, 6) in the contact frame and the corresponding Jacobian matrix and N_Γ is number of contact pairs. $\boldsymbol{\Gamma}(\mathbf{q}, \dot{\mathbf{q}})$ acts on the environment in addition to the forces due to the contact mechanics from MuJoCo (*i.e.* $\boldsymbol{\Omega}$).

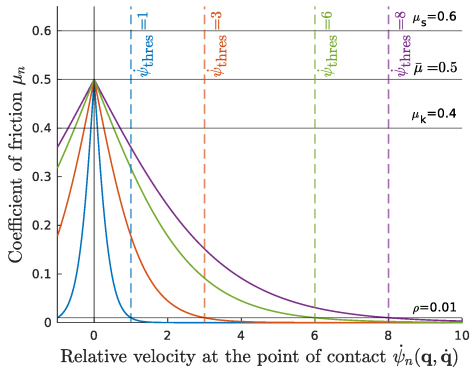

Fig. 2: The tunable smooth (except at $\dot{\psi}_n(\mathbf{q},\dot{\mathbf{q}})=0$) contact friction model that supplies virtual frictional force to break static friction in trajectory optimization.

1) Tunable Smooth Contact Models: We propose two tunable smooth contact models that supply virtual force. The first one (Eq. 4) [17] models virtual contact normal force $\mathbf{\Gamma}_n^N(\mathbf{q},\dot{\mathbf{q}})$ using a linear combination of nonlinear springs and dampers that resists penetration into the environment.

$$\mathbf{\Gamma}_n^N(\mathbf{q},\dot{\mathbf{q}}) = k_n e^{-\alpha_k \psi(\mathbf{q})} + b_n \text{sig}(-\alpha_b \psi_n(\mathbf{q}))\dot{\psi}_n(\mathbf{q},\dot{\mathbf{q}}) \quad (4)$$

where $\psi_n(\mathbf{q})$ is the depth of penetration, $\dot{\psi}_n(\mathbf{q},\dot{\mathbf{q}})$ is the relative pre-impact velocity at the point of contact, k_n is the contact stiffness or spring stiffness, b_n is the contact damping constant. Equation 5 models the virtual contact friction coefficient μ_n as a function of relative impact velocity at the point of contact (Fig. 2).

$$\mu_n = \bar{\mu} - \left| \frac{2\bar{\mu}}{1+\exp\left(\frac{\dot{\psi}_n(\mathbf{q},\dot{\mathbf{q}})}{\alpha_\mu}\right)} - \bar{\mu} \right|, \quad \alpha_\mu = \frac{\dot{\psi}_{\text{thres}}}{ln\left(\frac{\rho}{2\bar{\mu}-\rho}\right)} \quad (5)$$

where $\bar{\mu} = (\mu_s + \mu_k)/2$, μ_s and μ_k are the static and kinetic friction coefficients, μ_n is the virtual coefficient of friction, $\dot{\psi}_{\text{thres}}$ is the velocity threshold that breaks stiction and $\rho \to 0_+$ is a very small positive value. Then the virtual contact frictional force $\mathbf{\Gamma}_n^f(\mathbf{q},\dot{\mathbf{q}})$ is given as

$$\mathbf{\Gamma}_n^f(\mathbf{q},\dot{\mathbf{q}}) = \mu_n(\mathbf{\Gamma}_n^N(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{\Omega}^N) \quad (6)$$

Note that in Eq. 4 the normal force $\mathbf{\Gamma}_n^N$ is nonzero and acts from a distance (*i.e.* $\psi(\mathbf{q},\dot{\mathbf{q}}) > 0$) when $k_n, b_n \neq 0$. By using this virtual contact force, the optimizer can discover the contact locations to brace the robot on the environment and offset the torque limits of the robot. Similarly, the coefficient of friction is equal to the average of static and kinetic friction coefficients when $\dot{\psi}_n(\mathbf{q},\dot{\mathbf{q}}) = 0$. And based on the model parameter $\dot{\psi}_{\text{thres}}$ at which the object breaks static friction and starts sliding, the coefficient of virtual friction is equal to the very small value ρ and the frictional force from the physics engine takes over. This enables the opposing virtual friction to counteract the static friction from the physics engine and automatically discover sliding between the objects.

The trajectory optimization is set up with costs on the tunable parameters of the smooth virtual force models such that it minimizes the deviation from strict rigid body contact conditions (Sec. V-B). The net virtual force acting on a free body is the sum of the virtual forces associated with the contact candidates on that body, $n \in \{0, 1, \ldots, N_\Gamma\}$.

V. Torque-Limited Planning With Contact

Our planning framework interleaves graph search with trajectory optimization to combine the benefits of former's ability to search non-convex spaces and solve combinatorial parts of the problem and the latter's ability to obtain a locally optimal solution not constrained to discretization. We will first describe the graph search set up in the low-D space, and then the trajectory optimization in the full-D space that finds the control input trajectory along with the contact model parameters (Fig. 4). We will then explain how INSAT [1] is adapted for the application of torque-limited manipulation planning with contact. Finally, we provide experimental evidence (Sec. VI) that INSAT is superior in terms of the solution quality and the planner's behavior than the naive option of running them in sequence.

Consider an invertible many-to-one mapping $\boldsymbol{\lambda} : \mathcal{X} \longrightarrow \mathcal{X}_L$ that projects a full dimensional state $\mathbf{x} = [\mathbf{q},\dot{\mathbf{q}}] \in \mathcal{X}$ into the low-dimensional space $\mathcal{X}_L$. So $\mathbf{x}_L = \boldsymbol{\lambda}(\mathbf{x})$. Then $\boldsymbol{\lambda}^{-1} : \mathcal{X}_L \longrightarrow \mathcal{X}$ is an one-to-many inverse mapping of $\boldsymbol{\lambda}$ that lifts a low dimensional state $\mathbf{x}_L \in \mathcal{X}_L$ to any possible full-dimensional state $\mathbf{x} \in \mathcal{X}$. So $\mathbf{x} = \boldsymbol{\lambda}^{-1}(\mathbf{x}_L)$. $\boldsymbol{\phi}_{\mathbf{x}'\mathbf{x}''}(t)$ denotes a time t parameterized full-D trajectory from $\boldsymbol{\lambda}^{-1}(\mathbf{x}'_L)$ to $\boldsymbol{\lambda}^{-1}(\mathbf{x}''_L)$. The argument t is dropped for brevity.

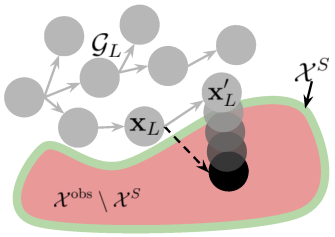

Fig. 3: Generation of contact configurations. When the low-D state $\mathbf{x}_L$ is expanded, the newly generated state is in collision (shown in black with a dashed edge) with $\mathcal{X}^{obs}$. In this work, we use the inbuilt property of MuJoCo to naturally repel the objects in collision to generate the first configuration $\mathbf{x}'_L$ that exits from $\mathcal{X}^{\text{obs}} \setminus \mathcal{X}^S$ to $\mathcal{X}^S$ as a successor.

A. Low-Dimensional Graph Search

To plan a trajectory that respects system dynamics and controller saturation, and simultaneously reason globally over large non-convex environments, it is imperative to maintain the combinatorial graph search tractable. To this end, we consider a low-dimensional space $\mathcal{X}_L$ (N-D) comprising of the joint angles $\mathbf{q}$. We build the low-D graph $\mathcal{G}_L$ by discretizing the free joint configuration space of the manipulator $(\mathcal{X} \setminus \mathcal{X}^{\text{obs}}) \cup \mathcal{X}^S$. Each edge in the graph corresponds to the robot's unit joint movement by a known distance (Fig. 4). Every newly generated node is checked to not violate joint angle limits and joint torque limits by calculating the gravity compensation before adding to the graph. So for an N DoF manipulator, the branching factor of the graph is $2N$ (unit joint movement in either direction satisfying joint angle and static joint torque limits). The graph search can be sped up using a heuristic $h(\mathbf{x}_L)$, an underestimate on the cost-to-goal of the optimal trajectory. We use the Euclidean distance between two nodes in the joint configuration space as our heuristic $h(\mathbf{x}_L) = \left\|\mathbf{x}_L - \mathbf{x}_L^G\right\|$

1) Contact vs. Collision: In motion planning, the task is to find a collision-free path from start to goal. This means making contact or touching the obstacle is considered as collision with the environment. However, for planning with

Algorithm 1 INSAT

1: **procedure** KEY($\mathbf{x}_L$)
2: **return** $g(\mathbf{x}_L) + \epsilon * h(\mathbf{x}_L)$
3: **procedure** GENERATETRAJECTORY($\mathbf{x}_L$, $\mathbf{x}'_L$)
4: **for** $\mathbf{x}''_L \in Ancestors(\mathbf{x}_L)$ **do** ▷ From $\mathbf{x}^S_L$ to $\mathbf{x}_L$
5: **if** $\mathbf{x}''_L = \mathbf{x}^S_L$ **then**
6: $\lambda^{-1}(\mathbf{x}''_L) = \mathbf{x}^S$
7: **else if** $\mathbf{x}'_L = \mathbf{x}^G_L$ **then**
8: $\lambda^{-1}(\mathbf{x}'_L) = \mathbf{x}^G$
9: $\phi_{\mathbf{x}''\mathbf{x}'} = O(\lambda^{-1}(\mathbf{x}''_L), \lambda^{-1}(\mathbf{x}'_L))$ ▷ Eq. 8
10: **if** $\phi_{\mathbf{x}''\mathbf{x}'}.IsCollisionFree()$ **then** ▷ Sec. V-A.1
11: $\phi_{\mathbf{x}^S\mathbf{x}'} = O_w(\phi_{\mathbf{x}^S\mathbf{x}''}, \phi_{\mathbf{x}''\mathbf{x}'})$ ▷ Eq. 8 with warm-start
12: **return** $\phi_{\mathbf{x}^S\mathbf{x}'}$
13: **return** NULL w/ discrete ∞ cost
14: **procedure** MAIN($\mathbf{x}^S, \mathbf{x}^G$)
15: $\mathbf{x}^S_L = \lambda(\mathbf{x}^S)$; $\forall \mathbf{x}_L, g(\mathbf{x}_L) = \infty$; $g(\mathbf{x}^S_L) = 0$
16: Insert $\mathbf{x}^S_L$ in OPEN with KEY($\mathbf{x}^S_L$)
17: **while** KEY($\mathbf{x}^G_L$) $= \infty$ **do** ▷ $\mathbf{x}^G_L = \lambda(\mathbf{x}^G)$
18: $\mathbf{x}_L$ = OPEN.$pop()$
19: **for** $\mathbf{x}'_L \in Succ(\mathbf{x}_L)$ **do**
20: $\mathbf{x}'_L = SoftCopy(\mathbf{s}_L)$
21: **if** $\mathbf{x}'_L \in$ CLOSED **then**
22: $\mathbf{x}'_L = DeepCopy(\mathbf{s}_L)$; $g(\mathbf{x}'_L) = \infty$
23: $\phi_{\mathbf{x}^S\mathbf{x}'}$ = GENERATETRAJECTORY($\mathbf{x}_L$, $\mathbf{x}'_L$)
24: **if** $J_{total}(\phi_{\mathbf{x}^S\mathbf{x}'}) < g(\mathbf{x}'_L)$ **then** ▷ Eq. 1
25: $g(\mathbf{x}'_L) = J_{total}(\phi_{\mathbf{x}^S\mathbf{x}'})$ ▷ Eq. 1
26: Insert $(\mathbf{x}'_L, \phi_{\mathbf{x}^S\mathbf{x}'})$ in OPEN with KEY($\mathbf{x}'_L$)

contact, the planner should be allowed to collide (or make contact) with the environment to leverage contact forces and offset robot's limits. To that end, we distinguish contact from collision by defining an obstacle surface $\mathcal{X}^S$. The obstacle surface is a subspace of the obstacle space such that the distance from any point in the obstacle to the free space is bounded by $\beta \to 0_+$.

$$\mathcal{X}^S = \{\mathbf{x} \in \mathcal{X}^{\text{obs}} \mid \left\|\mathbf{x} - \mathbf{x}^{\text{free}}\right\| < \beta,\ \mathbf{x}^{\text{free}} \in (\mathcal{X} \setminus \mathcal{X}^{\text{obs}})\} \quad (7)$$

When generating successors in the low-D graph (Fig. 3), the newly generated successor that is in collision is projected out of the obstacle space by using the intrinsic property of MuJoCo to repel intersecting rigid bodies to generate a state that first exits $\mathcal{X}^{\text{obs}} \setminus \mathcal{X}^S$. Such a state typically lies in $\mathcal{X}^S$ and forms the contact configuration.

B. Trajectory Optimization for Planning through Contact

The trajectory optimizer is set up to solve a boundary value problem by finding a joint torque input trajectory that connects the full-D subspaces $\lambda^{-1}(\mathbf{x}'_L)$ and $\lambda^{-1}(\mathbf{x}''_L)$ of two manipulator configurations $\mathbf{x}'_L$ and $\mathbf{x}''_L$. We solve this using Successive Convexification (SCvx) [12]. SCvx solves a sequence of smooth quadratic approximations of the original nonlinear problem subjected to linearized dynamics. But, as the manipulator dynamics with contact is discontinuous, the linearization of dynamics is poor. To alleviate this, we use the tunable soft contact model (Sec IV-.1) to solve the trajectory optimization problem (Eq. 8). We begin with the relaxed setting for the contact model (*i.e.* large values for $\mathbf{k} = [k_1, k_2, \ldots, k_{N_\Gamma}]^\mathsf{T}, \mathbf{b} = [b_1, b_2, \ldots, b_{N_\Gamma}]^\mathsf{T}, \boldsymbol{\mu} = [\mu_1, \mu_2, \ldots, \mu_{N_\Gamma}]^\mathsf{T}$ that correspond to non-zero virtual contact forces when not in contact and nonzero virtual frictional force when the object is at rest) in which the system dynamics (manipulator dynamics + contact dynamics) and its gradients are smooth and solve the Eq. 8 using SCvx.

$$\min_{\mathbf{u}[.],\mathbf{k},\mathbf{b},\boldsymbol{\mu}} \quad \left\|\mathbf{x}_L[N] - \mathbf{x}^G_L\right\| + \sum_{i=0}^{N-1}(\|\mathbf{u}[i]\| + \|\dot{\mathbf{x}}[i]\|) + \sum_{n=1}^{N_\Gamma}\|\mathbf{k}\| + \|\mathbf{b}\| + \|\boldsymbol{\mu}\| \quad (8a)$$

$$\text{s.t.} \quad \mathbf{x}[0] = \mathbf{x}^0; \mathbf{x}[i+1] = \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i) \quad (8b)$$

$$|\dot{\mathbf{x}}(t)| \le \dot{\mathbf{x}}_{\text{lim}}; |\ddot{\mathbf{x}}(t)| \le \ddot{\mathbf{x}}_{\text{lim}}; |\mathbf{u}(t)| \le \mathbf{u}_{\text{lim}} \quad (8c)$$

Note that the virtual forces disappear when $\mathbf{k}, \mathbf{b}, \boldsymbol{\mu} = \mathbf{0}$ (Eq. 4, 6) and the above minimization problem optimizes for that. Although Eq. 8 looks like direct shooting, the variant of SCvx combines the benefit of shooting and direct transcription by exploiting the sparsity in linear dynamics constraint during the convexification phase and maintaining dynamic consistency by rolling out the trajectory with the full nonlinear dynamics using MuJoCo (see [12]).

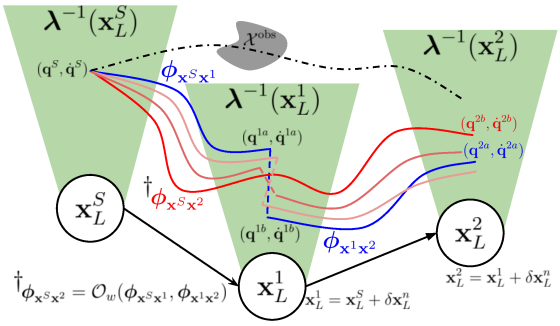

Fig. 4: Illustration of low-D graph, full-D subspaces of low-D states, trajectory optimization $O(.)$ and warm-started trajectory optimization $O_w(.)$ and iterating over low-D ancestors (line 4).

C. INSAT: INterleaved Search And Trajectory Optimization

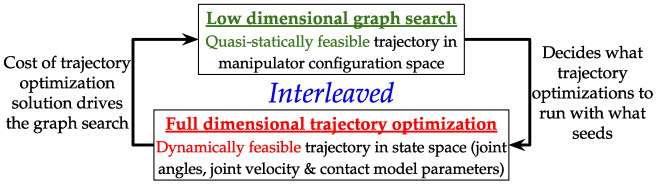

Fig. 5: A schematic of the working principle of INSAT

An overview of the algorithm is presented using a flowchart in Fig. 5. INSAT performs interleaved search on discrete low-dimensional manipulator configuration space and continuous high-dimensional joint velocity and contact model parameter space. The low dimensional search gets the manipulator around obstacles and evaluates various contact mode sequences. The high-dimensional trajectory optimization validates or invalidates the dynamic feasibility of paths discovered by the low-dimensional search. Consequently, INSAT generates dynamically feasible trajectories for the manipulator to brace with the environment, offset/stay within its torque limits, and reach the desired goal.

Alg. 1 presents the pseudocode of INSAT for torque-limited manipulation planning with contact. The algorithm takes as input the full-D start and goal states $\mathbf{x}^S$ and $\mathbf{x}^G$.

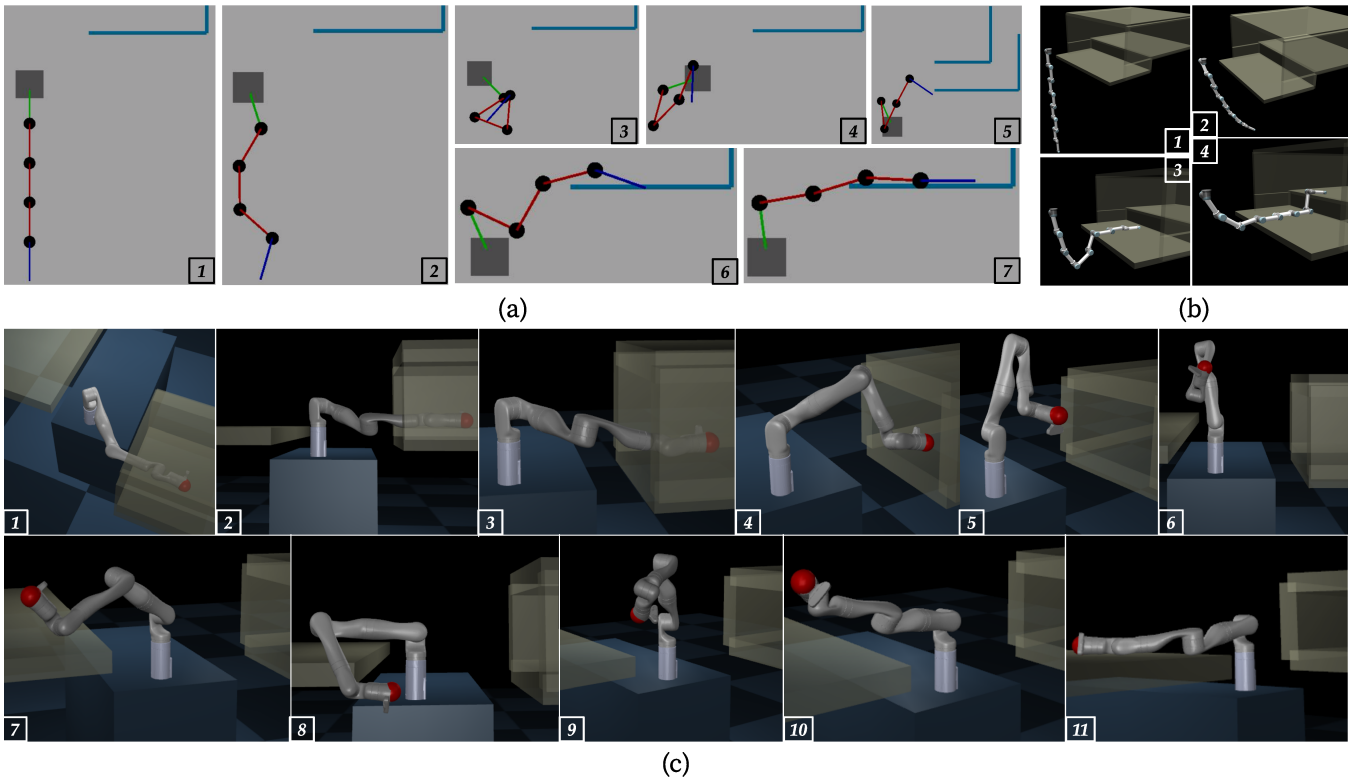

Fig. 6: Simulation experiments on (a) a planar arm, (b) hyper-redundant arm and (c) Kinova Gen3. In (a), the planar arm swings its way up into the ledge. It first rolls itself into a compact configuration (a3) to minimize the net torque by minimizing the moment arm and unfolds with the accrued momentum (a4) into the ledge. Once on the ledge, it slides its way by staying in contact and expending least joint effort. In (b), similar behavior as (a) is exhibited but on a 9 DoF redundant system. The robot has to slide and climb the shelf in a stretched out configuration. The robot climbs the stair without breaking contact. In (c) the Kinova Gen3 robot picks and places an overweight object (shown in red) from a confined shelf to a table. The robot's payload limit is 4kg and we used a 4.7kg payload. The mass violates the static torque limits without contact support from the environment at the start configuration. The robot maintains the contact with the shelf as much as possible by dragging the object out and swinging across its base to pump energy to eventually carry the object on to the table. The task requires reasonably long horizon planning in which standalone trajectory optimization struggles. By guiding the trajectory optimization with graph search over manipulator configurations, INSAT is able to produce a dynamically feasible trajectory of unique behavior.

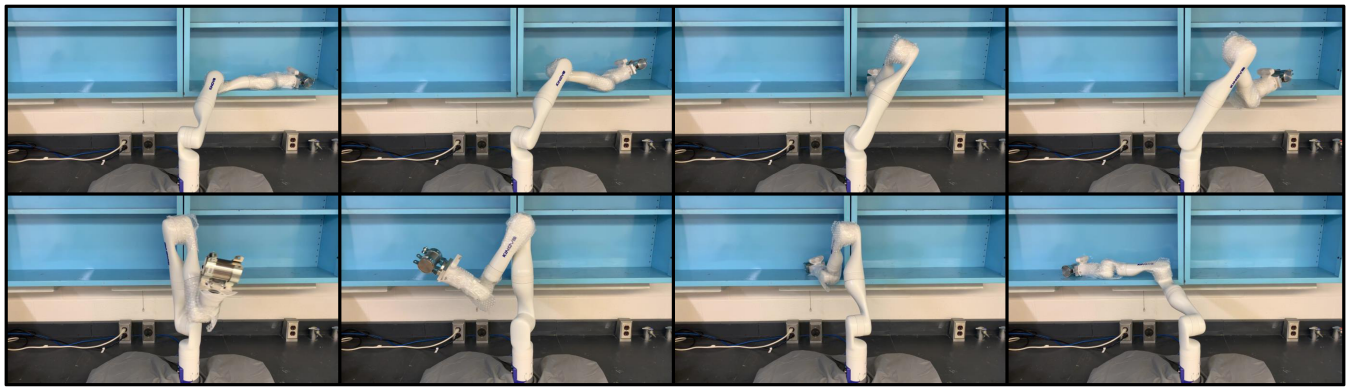

Fig. 7: Film strip showing a 7 DoF Kinova Gen3 robot utilizing bracing contacts to transfer a 2.5 kg payload between two cabinets using minimal torque. The arm slides the payload all the way to the center by bracing with its wrist before lifting on its own. The payload is then placed at the proximal end of the target shelf and pushed by bracing its forearm.

To search in the low-D graph $\mathcal{G}_L$, we use weighted A* (WA*)[25] which maintains a priority queue called OPEN that dictates the order of expansion of the states and the termination condition based on $\textsc{Key}(\mathbf{x}_L)$ value (lines 1, 17). Alg. 1 maintains two functions: cost-to-come $g(\mathbf{x}_L)$ and a heuristic $h(\mathbf{x}_L)$. $g(\mathbf{x}_L)$ is the cost of the current path from the $\mathbf{x}^S$ to $\mathbf{x}_L$ and $h(\mathbf{x}_L)$ is an underestimate of the cost of reaching the goal from $\mathbf{x}_L$. WA* initializes OPEN with $\mathbf{x}_L^S$ (line 16) and tracks the expanded states using another list called CLOSED (line 21).

A graphical illustration of how the low-D state expansions and full-D trajectory generations might look is shown in Fig 4. Each time the search expands a state $\mathbf{x}_L$, it removes $\mathbf{x}_L$ from OPEN and generates the successors as per the dis-

cretization (lines 18-20). For every low-dimensional successor $\mathbf{x}'_L$, we solve a trajectory optimization problem described in Sec. V-B to find a corresponding full-D trajectory from the farthest ancestor $\mathbf{x}''_L$ of $\mathbf{x}_L$ (line 4) to $\lambda^{-1}(\mathbf{x}'_L)$ (lines 9-11, Fig 4). The trajectory optimization output $\phi_{\mathbf{x}''\mathbf{x}'}$ is checked for collision (line 10, Sec. V-A.1). If the optimized trajectory $\phi_{\mathbf{x}''\mathbf{x}'}$ is in collision or infeasible (Fig 4), the algorithm continues with the next farthest ancestor (line 4). Upon finding the state $\mathbf{x}''_L$ which enables a full-dimensional feasible trajectory $\phi_{\mathbf{x}''\mathbf{x}'}$, the entire trajectory from start $\phi_{\mathbf{x}^S\mathbf{x}'}$ is constructed by warm-starting the optimization (O_w) with the trajectories $\phi_{\mathbf{x}^S\mathbf{x}''}$ and the newly generated trajectory $\phi_{\mathbf{x}''\mathbf{x}'}$ (line 11) by relaxing all the waypoint and derivative constraints (Fig. 4) until convergence or trajectory becoming infeasible, whichever occurs first.

VI. Experiments and Results

Before we present the results, we remark that our proposed method is the first algorithm that (i) introduces global, long horizon manipulation planning through bracing and (ii) demonstrates it on a physical robot or realistic robot examples in simulation. As such, we could not find a perfect baseline for comparison that solves our exact problem. So we used a method that is fairly common to generate smooth trajectories for manipulation planning. Also we intend to show the benefits of *interleaving* graph search and contact implicit trajectory optimization over the common choice of using them in *sequence* [20], [21], [22], [23]. To do so, we compare our method with a sequential combination of bi-directional RRT and direct collocation [23]. The choice of bi-directional RRT over graph search algorithms is that RRT variants are a popular choice for fast high-dimensional manipulation planning. All the methods are implemented in C++ on a 3.6GHz Intel Xeon machine.

A. Simulation Experiments

We show the results from simulation in three different environments and robot types namely (a) a planar 5-link arm climbing a ledge (b) a contrived hyper-redundant (9 DoF) version of UR5 manipulator that has to enter a rectangular tube and crawl over a step and (c) a pick and place task of an overweight payload by Kinova Gen3 robot. Due to page limit, we only provide qualitative analysis for (a) and (b) (see (Fig. 6) caption). In order to minimize torque and spend the least effort by exploiting contact, the manipulator exhibits swinging behavior to reach the contact locations and postures in all our simulation scenarios. Fig. 8 visualizes the planner's output torque trajectory for scenario (c) along with the torque reduction ratio (TRR) [2]. TRR = $(\|\boldsymbol{\tau}_{\text{wo}}\| - \|\boldsymbol{\tau}_{\text{c}}\|)/\|\boldsymbol{\tau}_{\text{c}}\|$ where $\boldsymbol{\tau}_{\text{wo}}$ is the net joint torque from the baseline and $\boldsymbol{\tau}_{\text{c}}$ is the net torque from INSAT (Eq. 3). As our baseline cannot discover and exploit contact, the value of $\boldsymbol{\tau}_{\text{wo}}$ is higher which explains a high TRR of 0.78. We also found that INSAT leverages passive dynamics as much as possible. Note from Fig. 8 that actuators 3, 5, 6 and 7 remain shut off with zero torque throughout the *slide out* and *swing* phase and activate only during the *land* phase (Fig 6). This suggests that the planner took advantage of the passive dynamics to effortlessly slide out of the confined shelf just using the actuators 2 and 4.

B. Real Robot Experiments

In order to experimentally validate our method, we planned a trajectory for a Kinova Gen 3 robot to lift a 2.5 Kg payload between adjacent cabinets as shown in Fig. 7. The robot was commanded in joint velocity mode using ROS Melodic on Ubuntu 18.04. As a comparison point, the same trajectory was executed in free space (i.e., without the cabinets). During both experiments, the joint torque was recorded using the Kinova Gen3's integrated torque sensors. Table I shows the RMS of the sensed joint torques in both experiments. From these values, it is clear that our planner was able to utilize bracing contacts to meaningfully reduce the torque in most joints when compared to free-space motion.

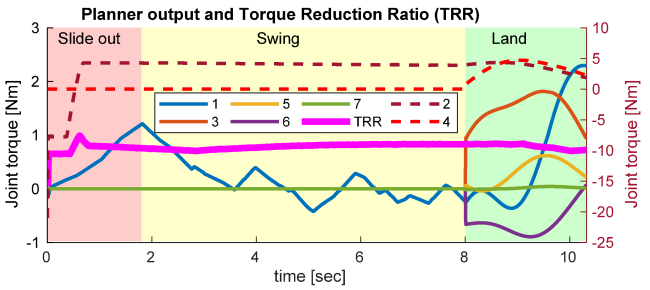

Fig. 8: Joint torque trajectory for simulation scenario (c) and TRR calculated with respect to our baseline (Bi-RRT + direct collocation). The torque plot for joints 2 and 4 using dotted lines use the y-axis on the right.

Joint ID	1	2	3	4	5	6	7	Total
Free-space	1.93	33.28	10.14	14.06	0.97	6.29	0.20	66.86
Bracing	4.65	23.96	6.67	11.80	1.28	5.67	0.59	54.62
Difference	-2.72	9.32	3.46	2.26	-0.31	0.63	-0.40	12.24

TABLE I: Experimental RMS torques [Nm] during (i) the braced trajectory shown in Fig. 7 and (ii) the same trajectory running in free-space (i.e. without the cabinets) producing net savings of 12.24Nm.

VII. Conclusion & Future Directions

In this work, we presented an interleaved approach to solving this problem with the aim of deploying it on real robots. This is achieved by interleaving graph search with continuous trajectory optimization. We show that planning with torque and obstacle constraints can be achieved in a way that finds bracing locations in the environment in order to make an otherwise inaccessible configuration reachable due to the torque reduction achieved by bracing. Experiments showed that the use of bracing contacts can reduce the required actuator torque for a given trajectory. The major limitations of this work are the planning time (taking in the order of 15-20 minutes for complex scenarios) and the fidelity of the dynamics model required to assess contact configurations for bracing. Our current work on the multi-threaded version of INSAT is showing around 15x reduction in planning time. Nevertheless, we believe that our algorithm can enable realistic deployment on robotic systems that operate in confined spaces. This will allow the next generation of minimal torque actuation robots to operate safely in deep and confined spaces for collaborative manufacturing. While in this work we utilize a full dynamics planner, some confined spaces may be too restricted to execute any dynamic behavior. Therefore, in future work, we will explore the minimum dynamic model fidelity needed for torque-limited manipulation through contact in confined spaces.

References

[1] R. Natarajan, H. Choset, and M. Likhachev, "Interleaving graph search and trajectory optimization for aggressive quadrotor flight," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5357–5364, 2021.

[2] C. Fang, N. Kashiri, G. F. Rigano, A. Ajoudani, and N. G. Tsagarakis, "Exploitation of environment support contacts for manipulation effort reduction of a robot arm," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9502–9508.

[3] R. Hollis and R. Hammer, "Real and virtual coarse-fine robot bracing strategies for precision assembly," in *Proceedings 1992 IEEE International Conference on Robotics and Automation*. IEEE Comput. Soc. Press, pp. 767–774.

[4] G. Wang and M. Minami, "Modelling and control of hyper-redundancy mobile manipulator bracing multi-elbows for high accuracy/low-energy consumption," in *Proceedings of SICE Annual Conference 2010*. IEEE, 2010, pp. 2371–2376.

[5] G. L. Johnston, A. L. Orekhov, and N. Simaan, "Kinematic modeling and compliance modulation of redundant manipulators under bracing constraints," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 4709–4716.

[6] W. J. Book, S. Le, and V. Sangveraphunsiri, "Bracing strategy for robot operation," in *Theory and Practice of Robots and Manipulators*. Springer, 1985, pp. 179–185.

[7] A. Petrovskaya, J. Park, and O. Khatib, "Probabilistic estimation of whole body contacts for multi-contact robot control," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 568–573.

[8] J. Park and O. Khatib, "Robot multiple contact control," *Robotica*, vol. 26, no. 5, pp. 667–677, 2008.

[9] A. D. Marchese, R. Tedrake, and D. Rus, "Dynamics and trajectory optimization for a soft spatial fluidic elastomer manipulator," *The International Journal of Robotics Research*, vol. 35, no. 8, pp. 1000–1019, 2016.

[10] D. Berenson, S. S. Srinivasa, D. Ferguson, and J. J. Kuffner, "Manipulation planning on constraint manifolds," in *2009 IEEE international conference on robotics and automation*. IEEE, 2009, pp. 625–632.

[11] K. Yunt and C. Glocker, "Trajectory optimization of mechanical hybrid systems using sumt," in *9th IEEE International Workshop on Advanced Motion Control, 2006.* IEEE, 2006, pp. 665–671.

[12] A. Ö. Önol, P. Long, and T. Padır, "Contact-implicit trajectory optimization based on a variable smooth contact model and successive convexification," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2447–2453.

[13] W. Li and E. Todorov, "Iterative linear quadratic regulator design for nonlinear biological movement systems." in *ICINCO (1)*. Citeseer, 2004, pp. 222–229.

[14] D. E. Stewart and J. C. Trinkle, "An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction," *International Journal for Numerical Methods in Engineering*, vol. 39, no. 15, pp. 2673–2691, 1996.

[15] M. Anitescu and F. A. Potra, "Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems," *Nonlinear Dynamics*, vol. 14, no. 3, pp. 231–247, 1997.

[16] T. Marcucci, M. Gabiccini, and A. Artoni, "A two-stage trajectory optimization strategy for articulated bodies with unscheduled contact sequences," *IEEE Robotics and Automation Letters*, vol. 2, no. 1, pp. 104–111, 2016.

[17] M. Neunert, M. Stäuble, M. Giftthaler, C. D. Bellicoso, J. Carius, C. Gehring, M. Hutter, and J. Buchli, "Whole-body nonlinear model predictive control through contacts for quadrupeds," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1458–1465, 2018.

[18] I. Mordatch, E. Todorov, and Z. Popović, "Discovery of complex behaviors through contact-invariant optimization," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–8, 2012.

[19] I. Mordatch, Z. Popović, and E. Todorov, "Contact-invariant optimization for hand manipulation," in *Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation*, 2012, pp. 137–144.

[20] C. Chen, P. Culbertson, M. Lepert, M. Schwager, and J. Bohg, "Trajectotree: trajectory optimization meets tree search for planning multi-contact dexterous manipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 8262–8268.

[21] Y.-C. Lin, B. Ponton, L. Righetti, and D. Berenson, "Efficient humanoid contact planning using learned centroidal dynamics prediction," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 5280–5286.

[22] Y.-C. Lin, L. Righetti, and D. Berenson, "Robust humanoid contact planning with learned zero-and one-step capturability prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2451–2458, 2020.

[23] X. Cheng, E. Huang, Y. Hou, and M. T. Mason, "Contact mode guided sampling-based planning for quasistatic dexterous manipulation in 2d," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6520–6526.

[24] 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.

[25] I. Pohl, "Heuristic search viewed as path finding in a graph," *Artificial intelligence*, vol. 1, no. 3-4, pp. 193–204, 1970.