

# GOMP-ST: Grasp Optimized Motion Planning for Suction Transport

Yahav Avigal\*, Jeffrey Ichnowski\*, Max Yiye Cao, and Ken Goldberg

\* Equal contribution

University of California, Berkeley, USA

**Abstract.** Suction cup grasping is very common in industry, but moving too quickly can cause suction cups to detach, causing drops or damage. Maintaining a suction grasp throughout a high-speed motion requires balancing suction forces against inertial forces while the suction cups deform under strain. In this paper, we consider Grasp Optimized Motion Planning for Suction Transport (GOMP-ST), an algorithm that combines deep learning with optimization to decrease transport time while avoiding suction cup failure. GOMP-ST first repeatedly moves a physical robot, vacuum gripper, and a sample object, while measuring pressure with a solid-state sensor to learn critical failure conditions. Then, these are integrated as constraints on the accelerations at the end-effector into a time-optimizing motion planner. The resulting plans incorporate real-world effects such as suction cup deformation that are difficult to model analytically. In GOMP-ST, the learned constraint, modeled with a neural network, is linearized using Autograd and integrated into a sequential quadratic program optimization. In 420 experiments with a physical UR5 transporting objects ranging from 1.3 to 1.7 kg, we compare GOMP-ST to baseline optimizing motion planners. Results suggest that GOMP-ST can avoid suction cup failure while decreasing transport times from 16 % to 58 %. For code, video, and datasets, see <https://sites.google.com/view/gomp-st>

**Keywords:** Motion Planning · Manipulation · Optimization.

## 1 Introduction

Vacuum suction cup grasping, due to its ability to quickly hold and release a large variety of objects, is a common grasping modality for robots in industrial settings such as warehouses and logistics centers. With the recent rise in demand for robot pick-and-place operations, the speed of object transport is critical. However, suction grasps can fail if the object is transported too quickly. Determining the conditions where suction grasps fail is non-trivial due to the difficult-to-model deformations of suction cups under stress. Existing analytic models make simplifying assumptions, such as rigid suction cups [31] or quasi-static physics [26, 30, 10]. An alternative is to heuristically slow motions when objects are held in suction grasps.

In prior work, the Grasp-Optimized Motion Planner (GOMP [13]) leveraged an unconstrained degree of freedom (DoF) around the grasp axis to optimize pick-and-place motions for parallel jaw grippers. Grasp-Optimized Motion Planning for Fast Inertial

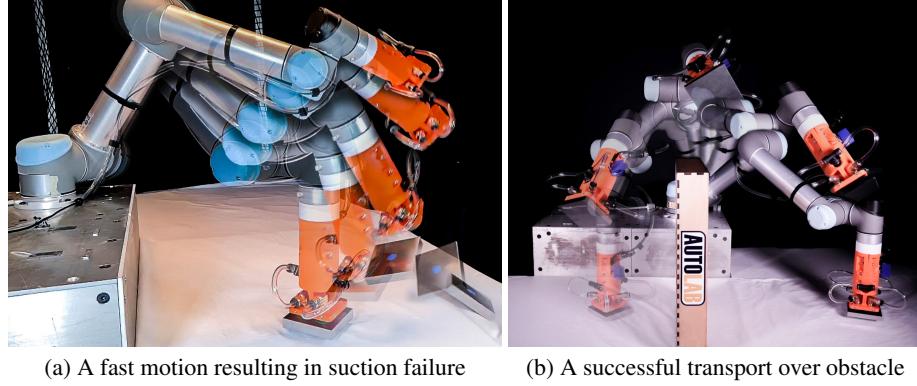


Fig. 1: **High-speed motions can cause failure of suction cups.** These multiple-exposure images show motions computed by the time-optimizing motion planners (a) J-GOMP and (b) GOMP-ST. The fast trajectory computed by J-GOMP and inertial forces cause the suction grasp to fail and the grasped block to fall away and to the right. We propose GOMP-ST, an algorithm that incorporates a learned acceleration constraint into a time-optimizing motion planner to avoid such failures.

Transport (GOMP-FIT [11]) computes time-optimized motions while taking into account end-effector and object acceleration constraints to reduce product damage and spills. However, when executed on robots with suction grippers, the resulting motions can fail due to suction cups detaching.

To address this problem, we propose the Grasp Optimized Motion Planning for Suction Transport (GOMP-ST), an algorithm that computes time-optimized motions by integrating a *learned* suction grasp loss constraint. GOMP-ST first tries varied rapid lifts of a given object with suction grasps to find motions that cause suction failures. It then learns a model from the data based on a history, or sequence, of end-effector accelerations to define a constraint function. At run time, the optimization treats the learned model as a non-linear constraint on the motion by linearizing it and computing a first-order approximation based on its Jacobian. In our implementation, we use a neural network to learn the suction model and use Autograd to obtain the Jacobian.

In experiments with steel rectangular blocks, we learn a model of suction failure on a physical UR5 with a 4-cup vacuum gripper. We then apply the learned model to transport 4 objects of varying mass held in suction between multiple different start and goal pairs and around obstacles. We compare GOMP-ST to GOMP, GOMP-FIT with an analytic model of suction, and ablations of GOMP-ST. We find that GOMP-ST can achieve a near 100 % success rate, for motions that are 16 to 58 % faster.

The contributions of this paper are:

- A novel algorithm, GOMP-ST, Grasp Optimized Motion Planning for Suction Transport, based on:
  1. Formulation of a learnable acceleration constraint for suction cup transport
  2. An efficient method for learning the constraint via boundary searching and data augmentation
  3. Integration of the learned constraint into an optimizing motion planner
- Data from experiments with a physical robot comparing GOMP-ST to baselines.

## 2 Related Work

### 2.1 Motion Planning and Optimization

Robot motion planning aims to find safe and efficient robot motions from a start to a goal configuration that avoid obstacles. Sampling-based motion planners, such as PRM [15] and bi-directional RRT [19] have variants that are probabilistically-complete and asymptotically-optimal. Optimization-based motion planners, such as TrajOpt [34], STOMP [14], CHOMP [32], KOMO [37], and ITOMP [29] can compute optimized trajectories iteratively improving paths or interleaving with sampling-based planners [18]. Grasp-Optimized Motion Planning (GOMP) [13] and Deep-Jerk GOMP (DJ-GOMP) [12] leverage a degree-of-freedom in the grasp pose while integrating dynamic and kinematic constraints in a sequential quadratic program that iteratively computes time-optimized pick-and-place motions. GOMP for Fast Inertial Transport (GOMP-FIT) [11] incorporates end-effector acceleration constraints for parallel-jaw grippers by employing the forward pass of the Recursive Newton Euler (RNE) method [22]. In contrast to prior work, GOMP-ST learns a suction-cup constraint to avoid suction grasp failure.

### 2.2 Constraint Learning in Optimization

Optimization using constraints based on empirical models is gaining traction. Maragno et al. [27] integrate learned constraints into a mixed-integer optimization using trust regions defined by the convex hull of the training data. In contrast, we propose using domain knowledge to perform data augmentation in a continuous optimization. Kudla et al. [17] propose a learning a decisions tree of constraints and a transform appropriate for mixed integer linear programming. While the decision tree provides flexible conditions for constraints, we instead propose exploiting the nature of the trajectory optimization by regressing on inputs containing the last  $h$  time steps. Bartolini et al. [2] and later Lombardi et al. [21] show that neural networks can learn constraints based on difficult-to-model phenomena be integrated into constrained combinatorial optimizations. We employ neural network constraints in continuous non-convex optimization. De Raedt et al. [6] provide a recent survey of constraint acquisition, and Fajemisin et al. [7] provide a recent survey of optimization with constraint learning. These surveys provide a wealth of ideas that could be extended to apply additional constraint learning and learned constraints to optimizing motion planning.

### 2.3 Suction Grasping

Suction grasping is widely used in industrial settings. Due to the increasing interest in suction grasping for pick-and-place tasks, recent models focus on computing a robust suction grasp. Dex-Net 3.0 [26] introduced a novel suction contact model that quantifies seal formation using a quasi-static spring system, along with a robust version of the model under random disturbing wrenches and perturbations in object pose, gripper pose, and friction. Huh et al. [10] use a learned model and a novel multi-chamber suction cup design to detect failures in the suction seal early to avoid grasp failures. However, these works assume quasi-static physics, an assumption that does not hold for high inertial forces.

Most closely related to this paper is the work by Pham and Pham [30] which proposes a suction-cup model and identify a contact stability constraint and a pipeline to parameterize time-optimal geometric paths satisfying the constraint. GOMP-ST differs in a few ways. First, their method plans a motion using a sampling-based planner, then time-parameterizes the motion, whereas GOMP-ST integrates planning and time-parameterization into a single optimization, allowing to explore alternate paths with potentially better timing. Second, their method uses an analytic model that does not include suction cup deformation, whereas GOMP-ST learns a constraint based on suction grasps failures through experimentation.

## 2.4 Dynamic Manipulation

Dynamic manipulation exploits forces due to accelerations, along with kinematics, static, and quasi-static forces to achieve a task [33]. Lynch and Mason [24] leverage centrifugal and Coriolis forces to allow low degree-of-freedom robots to control objects with more degrees-of-freedom. Lynch and Mason [25] also directly integrate constraints in a sequential quadratic problem to plan robot trajectories that achieve a dynamic task, such as snatching an object, throwing, and rolling, via coupling forces through the non-prehensile contact in an obstacle-free environment. Srinivasa et al. [35] address the problem of rolling a block resting on a flat palm by employing constraints on accelerations at the contact point. Mucchiani and Yim [28] propose a grasping approach that utilizes object inertia for sweeping an object at rest to a goal position by leveraging accelerations and torques from the path as stabilizing forces in the passive end-effector. In this work, we employ constraints on the end-effector accelerations to perform efficient dynamic object transport around obstacles using a suction gripper.

One promising related avenue of research investigates motion planning and optimization that integrates forces present at the end-effector. Hauser [9] investigates including contact forces during the optimization, and Luo and Hauser [23] extend this to include a learned confidence into an optimization. In a similar manner, Bernheisel and Lynch [3] and Acharya et al. [1] both explore different ways to address the *waiter’s problem* which requires generating motions of a tray to perform non-prehensile balancing of objects. In contrast to these lines of work, we focus on learning and integrating a constraint to maintain suction contact as a differentiable function, instead of learning the parameters of a model and integrating it into a plan.

When the dynamics are unknown or the existing models are insufficiently accurate, a promising approach is to leverage data-driven methods. Zeng et al. [40] propose TossingBot that learns parameters of a pre-defined dynamic motion to toss objects into target bins using parallel-jaw end-effectors. Wang et al. [39] propose SwingBot, that uses tactile feedback to learn how to dynamically swing up novel objects. In Robots of the Lost Arc [41], a robot computes high-speed motions to induce fixed-end cables to swing over distant obstacles, knock down target objects, and weave between obstacles. Ha and Song [8] propose FlingBot, a self-supervised learning framework to learn dynamic flinging actions for cloth unfolding. Lim et al. [20], uses simulation and physical data to train a model for swinging an object to hit a target. In this work, we integrate a learned constraint in the solver allowing it to compute fast motions while maintaining the suction seal.

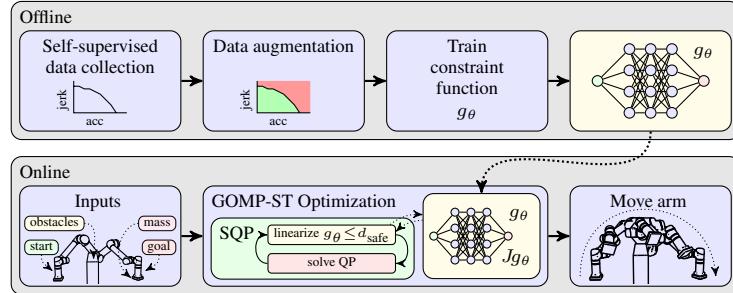


Fig. 2: **GOMP-ST pipeline.** In an offline process (**top**), GOMP-ST performs self-supervised learning of a constraint function. It first repeatedly grasps objects of known mass to perform a boundary search on a motion profile parameterized by jerk and acceleration limits. Then, during data augmentation, it labels slower motions as grasp successes, and faster motions as grasp failures. Finally, it trains a neural network constraint function  $g_\theta$ . In the online process (**bottom**), GOMP-ST computes a motion plan for a given problem. The SQP solver repeatedly linearizes the learned suction constraint using a user-specified threshold  $d_{\text{safe}} \in [0, 1]$  and the output and Jacobian (via autograd) of the trained neural network.

### 3 Problem Statement

Let  $\mathbf{q} \in \mathcal{C}$  be the complete specification of the degrees of freedom for a robot, where  $\mathcal{C}$  is the set of all possible configurations. Let  $\mathcal{O}$  be the set of obstacles, and  $\mathcal{C}_{\text{obs}} \subseteq \mathcal{C}$  be the set of configurations in collision with  $\mathcal{O}$ . Let  $\mathcal{C}_{\text{free}} = \mathcal{C} \setminus \mathcal{C}_{\text{obs}}$  be the set of configurations that are not in collision. Let  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  be the first and second derivatives of the configuration. Let  $\mathbf{a}_{\text{ee}} \in \mathbb{R}^3$  be the linear acceleration of the end effector with one or more suction cups holding a known object  $b$ , and  $\hat{\mathbf{n}} \in \mathbb{R}^3$  be the suction cup normal, as shown in Fig. 4.

Given a start  $\mathbf{q}_0$  and goal  $\mathbf{q}_{\text{goal}}$  configuration, the objective of GOMP-ST is to compute a trajectory  $\tau = (\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_H)$ , where  $\mathbf{x}_t = (\mathbf{q}_t, \dot{\mathbf{q}}_t, \ddot{\mathbf{q}}_t) \in \mathcal{X}$  is the state of the robot at time  $t$ ,  $\mathcal{X}$  is the set of states, such that  $\mathbf{q}_t \in \mathcal{C}_{\text{free}} \forall t \in [0, H]$ , and  $\dot{\mathbf{q}}$  and  $\ddot{\mathbf{q}}$  are within the box-bounded dynamic limits of the robot, and  $\mathbf{q}_H = \mathbf{q}_{\text{goal}}$ . The object  $b$  will remain attached throughout the motion. As with GOMP, start and goal configurations may also be expressed via forward kinematics and bounds on degrees of freedom.

### 4 Method

GOMP-ST learns a suction cup acceleration constraint via a sequence of physical experiments, then integrates it with a time-optimizing motion planner. See Fig. 2 for an overview. This section starts with background on the prior work, then describes how to learn and integrate the constraint into the motion planner.

#### 4.1 Background: GOMP-FIT

The GOMP-FIT [11] algorithm formulates a fast-inertial-transport motion planning as an optimization problem and solves it with sequential quadratic program (SQP) trust-

region-based solver. It first discretizes the trajectory into a sequence of  $H + 1$  waypoints ( $\mathbf{x}_0, \dots, \mathbf{x}_H$ ) that are each separated by a fixed time interval  $t_{\text{step}}$ . Each waypoint  $\mathbf{x}_t$  includes the configuration and its first and second derivatives  $\mathbf{x}_t = (\mathbf{q}_t, \dot{\mathbf{q}}_t, \ddot{\mathbf{q}}_t)$ . The outer loop shrinks  $H$  to find a minimum time trajectory. The inner loop solves an SQP where the optimization objective minimizes the sum-of-squared accelerations, the linear constraints keep the motion within the kinematic and dynamics limits of the robot, and the non-linear constraints avoid obstacles and limit linear accelerations experienced at the end-effector. To limit shock, it constrains the magnitude of end-effector acceleration to be below a threshold  $a_{\text{safe}}$ . To avoid spills, it constrains the angle between the container normal and the acceleration to be below a spill threshold  $\theta_{\text{spill}}$ . The optimization is

$$\begin{aligned} \min_{\mathbf{x}_{[0..H]}} \quad & \frac{1}{2} \sum_{t=0}^H \ddot{\mathbf{q}}_t \\ \text{s.t.} \quad & \mathbf{x}_{\min} \leq \mathbf{x}_t \leq \mathbf{x}_{\max} \quad \forall t \in [0..H] \\ & \mathbf{q}_{t+1} = \mathbf{q}_t + \dot{\mathbf{q}}_t t_{\text{step}} + \frac{1}{2} \ddot{\mathbf{q}}_t t_{\text{step}}^2 \quad \forall t \in [0..H) \\ & \dot{\mathbf{q}}_{t+1} = \dot{\mathbf{q}}_t + \ddot{\mathbf{q}}_t t_{\text{step}} \quad \forall t \in [0..H) \\ & f_{\mathcal{O}}(\mathbf{q}) \geq 0 \quad \forall t \in [0..H] \\ & \cos^{-1}(f_a(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \cdot f_n) \leq \theta_{\text{spill}} \quad \forall t \in [0..H] \\ & \|f_a(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})\| \leq a_{\text{safe}} \quad \forall t \in [0..H], \end{aligned}$$

where  $f_{\mathcal{O}} : \mathcal{C} \rightarrow \mathbb{R}$  is the signed distance from robot to set  $\mathcal{O}$  (thus implementing the constraint  $\mathbf{q}_t \in \mathcal{C}_{\text{free}}$ ) and  $f_a : (\mathcal{C})^3 \rightarrow \mathbb{R}^3$  is the linear acceleration at the end-effector computed using the Recursive Newton-Euler (RNE) algorithm. Additionally, GOMP-FIT optionally integrates constraints to optimize the grasp angle and location.

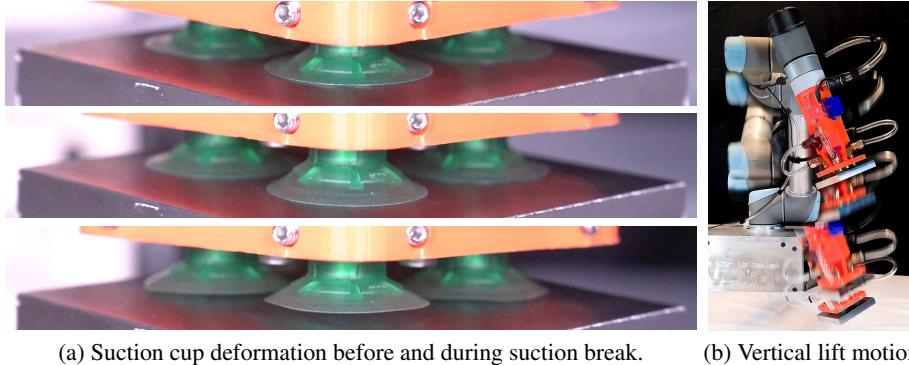
The SQP solver repeatedly linearizes the non-linear constraints (collision, end-effector acceleration, and grasp) to form a quadratic program (QP), and solves the QP, accepting solutions that improve the trajectory. In this optimization, when solving for the  $(k + 1)$  iterate, a non-linear constraint of the form  $g(\mathbf{x}) \leq y$  is linearized around the current iterate  $\mathbf{x}^{(k)}$  via a first-order approximation using its Jacobian:

$$J\mathbf{x}^{(k+1)} \leq y - g(\mathbf{x}^{(k)}) + J\mathbf{x}^{(k)}.$$

GOMP-FIT optimizes the trajectory time by repeatedly solving the SQP with a shrinking horizon  $H$ , warm-starting each subsequent SQP solve with an interpolation of the solution from the prior horizon. The optimization terminates the smallest  $H$  the solver detects as feasible. However, when executed on robots with suction grippers, the resulting motions can lead to suction failures.

## 4.2 Learned constraints in the SQP

Maintaining a suction grasp throughout a high-speed motion requires balancing suction forces against inertial forces while the suction cups deform under strain. GOMP-ST defines a series of physical experiments to learn a constraint from real-world data. We



(a) Suction cup deformation before and during suction break. (b) Vertical lift motion

Fig. 3: **Suction cup deformation observed with data collection.** (a) These frames from a slow-motion video show the deformation of suction cups as the gripper is rapidly pulled upward. At the beginning of the motion ((a) **top**), the suction cups compress against the grasped surface. As the gripper lifts and starts to break from suction ((a) **middle**), the suction cups deform but still maintain a seal. Continuing to pull away results in a suction grasp failure ((a) **bottom**). (b) During data collection, the robot lifts the mass with a vertical motion while grasping at an angle.

model the learned constraint as a function  $g_\theta : (\mathcal{X})^h \rightarrow [0, 1]$  parameterized by  $\theta$ , where  $(\mathcal{X})^h$  is history of  $h$  dynamic states, and a value of 0 indicates the object is held, while a value of 1 indicates a failure. (Notationally, we use  $h$  to indicate a history of states, and  $H$  to indicate the total trajectory length.)

We integrate this function into the optimization as a non-linear constraint:

$$g_\theta(\cdot) \leq d_{\text{safe}},$$

where the argument is a portion of the state from of the trajectory being optimized, and  $d_{\text{safe}} \in [0, 1]$  is a tunable failure threshold. To linearize this constraint for the SQP, we use the automatic gradients (Autograd) provided by a neural-network package [5] to compute the Jacobian.

Using slow-motion video capture, we observe that suction cups deform before suction failure (Fig. 3), and thus failures are not an instantaneous response to a change in end-effector state. With this observation, we propose that  $g_\theta$  should be a function of a sequence (or history) of states of length  $h$ . The value of  $h$  depends on the geometry and material of the suction cup. In a series of experiments we set  $h$  to be long enough to capture the time between deformation start and suction failure that we observe from slow-motion video playback ( $> 0.1$  seconds). Further, to translate the optimization variables from the state of the robot to the state of the end-effector, we utilize RNE function  $f_a$  for each state in the history. Thus we formulate the full constraint as:

$$g_\theta(f_a(\mathbf{x}_{t-h+1}), \dots, f_a(\mathbf{x}_{t-1}), f_a(\mathbf{x}_t)) \leq d_{\text{safe}} \quad \forall t \in [1..H].$$

When  $\mathbf{x}$  has a negative subscript (i.e., it refers to a state before the start of planning), we replace it with  $\mathbf{x}_0$ .

### 4.3 Self-supervised data collection and training

To learn  $g_\theta$ , GOMP-ST implements a self-supervised pipeline that defines a series of experiments to lift objects of known mass and collect data. We attach a pressure sensor to the tube connected to the suction cups, similar to Huh et al. [10]. To minimize delay between pressure changes and pressure readings, we place the sensor close to the suction cups. During each lift, the pipeline records the joint state and pressure sensor readings over time.

To isolate gravity during data collection, the system always lifts against gravity while varying the angle of the suction normal (see Fig. 3). As convention,  $0^\circ$  is a top-down grasp and  $90^\circ$  is a grasp in which the suction normal is perpendicular to gravity.

The pipeline computes each lift motion using Ruckig [4] between two points in end-effector (Cartesian) coordinates, and uses an inverse kinematics solver to translate into joint configurations. Ruckig computes straight-line time-optimal motions subject to velocity, acceleration, and jerk limits. During data collection, GOMP-ST varies the motion profile by changing the maximum acceleration  $a_{\max}$  and maximum jerk  $j_{\max}$  parameters of Ruckig.

The pipeline defines a discretized grid with  $a_{\max}$  and  $j_{\max}$  axes to fill with values 0 or 1 labels. To reduce data collection time, GOMP-ST performs a boundary search on the motion profiles. It starts with a fixed lower value of  $a_{\max}$  and increases  $j_{\max}$  until it observes a change in the pressure measurements indicating a suction grasp failure. Afterwards, it iteratively decreases  $j_{\max}$  or increases  $a_{\max}$  so that it is always exploring the above and below the continuous boundary at which suction fails.

After a suction failure, the automated data collection pipeline takes a top-down image of the scene to find and re-grasp the target object. After confirming the grasp using the pressure measurements, GOMP-ST moves the object to a consistent starting pose before performing the next lift.

The pipeline then trains a multi-layer perceptron with exponential linear units (ELU) activations using the joint and pressure data it collected. We choose ELU for its continuous gradients. Training details are in the experiments section. The pipeline scans the recorded pressure data to find the time at which suction pressure is lost. It then tracks a tunable number of steps  $h$  back to create a labeled data point containing  $h$  accelerations that led to the suction failure (labelled as 1). All sequences of  $h$  accelerations prior to the suction failure, or in records without suction failures, the pipeline labels as 0. GOMP-ST further perform data augmentation by scaling accelerations leading to failures by  $1 + \epsilon$ , and non-failures by  $1 - \epsilon$ , for small positive values of  $\epsilon$ .

### 4.4 Analytic model of suction-cup failure for GOMP-FIT baseline

As a baseline to the learned model, we compare GOMP-ST to GOMP-FIT where its constraint on the magnitude and direction of the inertial acceleration vector is provided by an analytic model. In previous work, Kolluru et al. [16] and Stuart et al. [36] proposed analytic models for rigid suction cups, but were limited to either single suction cups or symmetrical systems. Our analysis is more closely related to the more general method used by Valencia et al. [38] and Pham and Pham [30] which generalises over multiple suction cups and asymmetric loads, though still with the assumption of rigid

bodies. We also choose the rigid-body assumption as opposed to for example the spring model suggested by Mahler et al. [26], since a spring model of suction cups where quasi-static equilibrium is not assumed would require knowledge about the state of the springs. Alternatively, if quasi-static equilibrium is assumed, the state of the springs may be estimated, but the purpose of the additional complexity is defeated from a motion planning perspective as the resulting constraint would be equivalent to that of a rigid-body model. In this work, the analytic model makes the following assumptions:

1. There are quasi-static conditions in inertial frame, enabling equilibrium analysis.
2. The suction cups are rigid and modelled as point contacts.
3. The transported object is a rigid rectangular cuboid with uniform mass distribution such that its center of mass corresponds to its centroid.
4. There are no air leaks between the suction cups and flat grasping surface, which results in a static and equal suction force across all suction cups.

Consider the free-body diagram in the inertial frame of the object shown in Fig. 4, where  $\mathbf{f}_{s,i}$  are the suction forces,  $\mathbf{f}_{n,i}$  are the contact forces between the suction cups and grasped object,  $\mathbf{f}_{f,i}$  are the friction forces,  $\mathbf{f}_g$  is the gravitational force, and  $\mathbf{a}$  is the acceleration from a balance of forces.  $\hat{\mathbf{n}}_{\text{obj}}$  is the unit length (denoted by the hat) normal defining the grasping plane, and  $\theta = \cos^{-1}[(\mathbf{a} \cdot \hat{\mathbf{n}}_{\text{obj}})/(\|\mathbf{a}\|_2 \|\hat{\mathbf{n}}_{\text{obj}}\|_2)]$  is the angle between the inertial acceleration vector and suction force normal. In the idealised condition,  $\mathbf{f}_{s,i} = (p_{\text{atm}} - p_v)A$ , where  $p_{\text{atm}} - p_v$  is the difference between atmospheric pressure  $p_{\text{atm}}$  and applied vacuum pressure  $p_v$ , and  $A$  is the effective area of grip for a single suction cup. Furthermore, under the assumption of dry Coulomb friction,  $\mathbf{f}_{f,i} \leq \mu \mathbf{f}_{n,i}$ , where  $\mu$  is the static coefficient of friction between the grasped object and suction cups. Finally,  $\mathbf{f}_g = mg$ , where  $m$  is the object mass,  $g$  is the gravitational acceleration.

We define 2 models, *unimodal* and *multimodal*. The unimodal model assumes  $\theta = 0^\circ$ , and thus the maximum inertial acceleration before grasp failure is trivially given by  $\mathbf{a}_{\text{fail}} = \frac{1}{m} \sum_i \mathbf{f}_{s,i}$ . The multimodal model includes multiple failure modes for cases where  $\theta \neq 0^\circ$ , including suction grasp failure by sliding, force imbalance, or moment imbalance. This model does not include deformation. The analysis uses the equations

$$\sum_i \mathbf{f}_{s,i} + \sum_i \mathbf{f}_{n,i} + \sum_i \mathbf{f}_{f,i} + \mathbf{f}_g = m\mathbf{a} \quad (1)$$

$$\sum_i \mathbf{r}_i \times \mathbf{f}_{s,i} + \sum_i \mathbf{r}_i \times \mathbf{f}_{n,i} + \sum_i \mathbf{r}_i \times \mathbf{f}_{f,i} = \mathbf{0} \quad (2)$$

$$\mathbf{f}_{n,i} = -\alpha_i \cdot \hat{\mathbf{n}}_{\text{obj}} \quad (3)$$

$$\hat{\mathbf{f}}_{f,i} = -\widehat{\text{proj}_{\hat{\mathbf{n}}_{\text{obj}}}}(\mathbf{f}_g) \quad (4)$$

$$\mathbf{f}_{f,i} = \beta_i \cdot \hat{\mathbf{f}}_{f,i} \quad (5)$$

$$(\mathbf{f}_{f,i} \cdot \hat{\mathbf{f}}_{f,i}) = \gamma \cdot \mathbf{f}_{n,i} \cdot (-\hat{\mathbf{n}}_{\text{obj}}). \quad (6)$$

The force (1) and moment (2) balance in quasi-static equilibrium, where  $\mathbf{r}_i$  is the position vector from the object center of mass to the application of forces point for the  $i$ th suction cup. We then require the reaction forces  $\mathbf{f}_{n,i}$  to be normal to the grasping plane (3), set the direction of the friction forces to be tangent in the grasping plane and

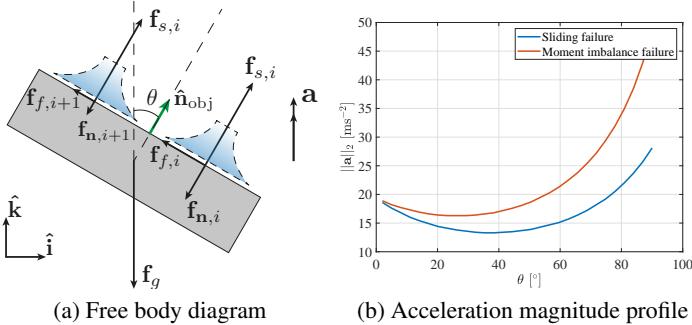


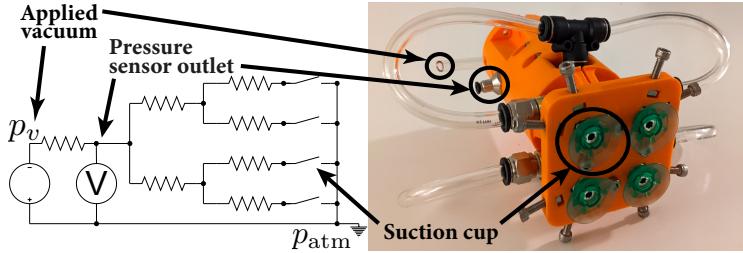
Fig. 4: **Analytic model of suction cup failure** (a): A set of suction cups (light blue) hold the gray object, where the suction surface normal is a vector  $\hat{\mathbf{n}}$  (not shown) that is always aligned with the vector  $\hat{\mathbf{n}}_{\text{obj}}$ . The suction forces  $\mathbf{f}_{s,i}$  apply the same suction force at each suction contact, while the normal forces  $\mathbf{f}_{n,i}$  react unevenly based on the tilt angle  $\theta$  relative to inertial acceleration and gravity  $\mathbf{f}_g$ . The green arrow indicates the unit normal direction vector, and all other arrows are force vectors. Under the unimodal model  $\theta = 0^\circ$ . Under the multimodal model  $\theta$  can vary. (b) shows the acceleration magnitude drop boundaries in the sliding and moment imbalance failure modes for varying angles  $\theta$  between the inertial acceleration  $\mathbf{a}$  and the suction normal  $\hat{\mathbf{n}}_{\text{obj}}$ . Suction fails for accelerations in the region above the blue curve in the model. We generate this plot by solving the system of equations (1) – (6) with a combination of angles and accelerations. For the parameters such as object dimensions and friction coefficient of our system, sliding will always occur before moment imbalance.

opposing the direction of the motion in its absence (4), (5), and require that the friction force magnitudes share the same proportionality constant  $\gamma$  to the normal forces (6).

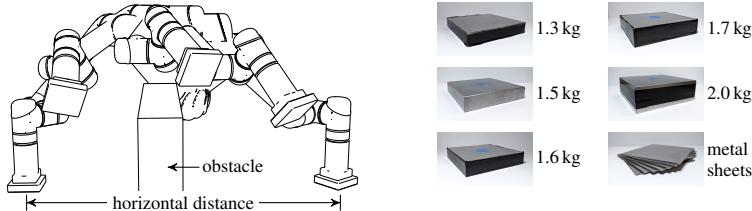
We solve the system of equations for a range of hypothetical angles  $\theta$  and accelerations  $\mathbf{a}$ , and classify each scenario as a failure or a success based on the physical restrictions that  $\beta_i \leq \mu\alpha_i$  and  $\alpha_i \geq 0$ , where  $\alpha_i$  and  $\beta_i$  are the magnitudes of the normal and frictional forces, respectively. The result of this simulation is shown in Fig. ??, which shows that the multimodal model converges with the unimodal model when  $\theta \rightarrow 0^+$ , and that in contrast to intuition, the curve is not strictly increasing nor strictly decreasing across the domain. The inputs used to generate the curve use dimensions from our experimental setup, and so the exact axes values do not generalise to other systems. To integrate the model with GOMP-FIT, we approximate the curve using a 4th-order polynomial fit, which we then use to formulate an analytic baseline constraint.

## 5 Experiments

We perform experiments on a physical UR5 with a custom 4-cup vacuum gripper and a set of steel blocks. The gripper has four round 30 mm diameter elastodur flat suction cups driven by a single VacMotion MSV 27 vacuum generator. Multi-cup suction grippers are common in many commercial automated logistics systems due to the increased surface area producing more suction force, and the multiple contact point stabilizing the hold resulting in reduced payload swing. The pressure sensor (Adafruit MPRLS Ported Pressure Sensor Breakout) is fitted in the gripper assembly and attached via USB to



**Fig. 5: Suction gripper with four suction cups and pressure sensor outlet.** In the custom suction gripper (**right**), the suction is distributed to the four suction cups using branching connectors from the pneumatic tube where a vacuum/negative pressure is applied, as shown in the analogous circuit diagram (**left**). A single pressure sensor is used to measure the applied pressure. In the case of a successful suction grasp, all four switches are open, and so no current/air flows, implying the applied pressure/voltage is equal for all four suction cup regardless of differing resistor values.



**Fig. 6: Experimental setup** In experiments, (**left**) we compute trajectories for varying horizontal transport distances, and average the results. Shorter distances require lifting faster, while longer distances can lift more gradually. We train and test on masses (**right**) composed of stacked steel blocks and sheets. We do not train on the 1.6-kg mass to test generalization to unseen masses.

the computer that drives the UR5. See Fig. 1 for a visual of the experimental setup and Fig. 5 for a close-up of the suction gripper.

We first perform data collection and train (Sec. 4.3) on four different masses, 1.343-, 1.492-, 1.741-, and 2.196-kg steel blocks (Fig. 6 right), which we round to 1.3, 1.5, 1.7, and 2 kg hereafter. We then test GOMP-ST and baselines by computing trajectories between varying start and goal positions and around varying obstacles. We automate data collection by using an overhead Intel RealSense 435i to locate the mass after suction failures. We use a relatively heavy mass to lower the end-effector accelerations required for a lost grasp, and for safety as a lost grasp results in the released object having lower kinetic energy  $E_k$ , since  $E_k = (1/2)mv^2$ , and thus do not fly out of the workspace.

We set the automated system to collect training data for each of the four training masses, resulting in 2,367 training trajectories. We perform an 80/20 train/test split on the trajectories, data-augment failures 30× and non-failures 4× in the training set. With the UR5 operating and generating joint data at a 125 Hz, and the pressure sensor operating at 167 Hz, this results in 597,010 training examples and 19,892 test examples.

With 15 trials each, we compare to baselines of J-GOMP, an optimizing motion planner that does not include inertial effects, and GOMP-FIT, an extension of J-GOMP that allows constraints on the linear acceleration at the end-effector. We include 2 GOMP-FIT baselines in which we limit magnitude of end-effector acceleration using the unimodal model and multimodal model.

**Table 1: Object transport success rate and motion time.** We compute 3 pick-and-place motions with 0.8, 0.9, and 1.0 m horizontal separation between pick and place points, and perform each motion 5 times, for a total of 15 trials per algorithm and mass. For algorithms, we use **J-GOMP** as a time-optimized, but not suction-constrained baseline to show a lower bound on time, **GOMP-FIT** with analytic constraints on end-effector acceleration, and **GOMP-ST** with varying history length  $h$ , and failure threshold  $d_{\text{safe}}$ . Masses 1.3, 1.5, and 1.7 were seen at train time; mass 1.6 was not. We highlight multimodal GOMP-FIT and GOMP-ST ( $h = 6, d_{\text{safe}} = 0.50$ ), and compare the relative speedup, observing that only 1 of the 60 trials for GOMP-ST failed, while GOMP-ST speeds up between 16 % and 58 %.

Mass [kg]	J-GOMP	GOMP-FIT		GOMP-ST ( $h, d_{\text{safe}}$ )			Speedup
		Unimodal	Multimodal	(1, 0.50)	(6, 0.05)	(6, 0.50)	
<b>Success Rate</b>							
1.3	0 %	0 %	<b>100 %</b>	0 %	<b>100 %</b>	<b>100 %</b>	0 %
1.5	0 %	0 %	<b>100 %</b>	0 %	<b>100 %</b>	<b>100 %</b>	66.7 %
1.6*	0 %	0 %	<b>100 %</b>	0 %	<b>100 %</b>	<b>93.3 %</b>	<b>100 %</b>
1.7	0 %	33.3 %	<b>100 %</b>	0 %	<b>100 %</b>	<b>100 %</b>	33.3 %
<b>Motion Time [s]</b>							
1.3	1.355	1.333	<b>2.304</b>	1.440	2.637	<b>1.931</b>	1.728 <b>+16 %</b>
1.5	1.355	1.387	<b>2.827</b>	1.525	2.763	<b>1.984</b>	1.781 <b>+30 %</b>
1.6*	1.355	1.472	<b>2.827</b>	1.643	2.795	<b>1.807</b>	1.781 <b>+36 %</b>
1.7	1.355	1.728	<b>4.459</b>	1.173	2.432	<b>1.856</b>	1.792 <b>+58 %</b>

\* mass unseen in training.

## 5.1 Ablation studies

We perform two ablation studies on GOMP-ST. First, we compare history length of  $h = 1$  to  $h = 6$ , to study the importance of the motion history relative to single acceleration spikes in the motion. We also compare different values of the failure threshold  $d_{\text{safe}}$  to study the potential for making a trade-off between speed and reliability.

## 5.2 Results

We vary the horizontal distance between start and goal configurations and compute trajectories for each baseline and variant of GOMP-ST (Fig. 6 left). Each trajectory lifts over an obstacle, thus the shorter horizontal distance (0.8 m) requires more vertical motion, while the longer horizontal distance (1.0 m) results in a period of longer horizontal acceleration. We also vary the masses, using 3 masses that were seen during training (1.3 kg, 1.5 kg and 1.7 kg) and a single mass that was not used during the data collection (1.613 kg, rounded to 1.6 kg). We run each computed trajectory 5 times and report the average success rate and motion time per transported mass as the experiment’s result.

In Table 1, we show the results of trajectories computed by the baselines and ablations. Here we see that the J-GOMP and the GOMP-FIT unimodal baselines consistently compute motions that lead to suction failures. GOMP-FIT multimodal manages to compute safe motions, however the resulting trajectories are slow. In contrast,

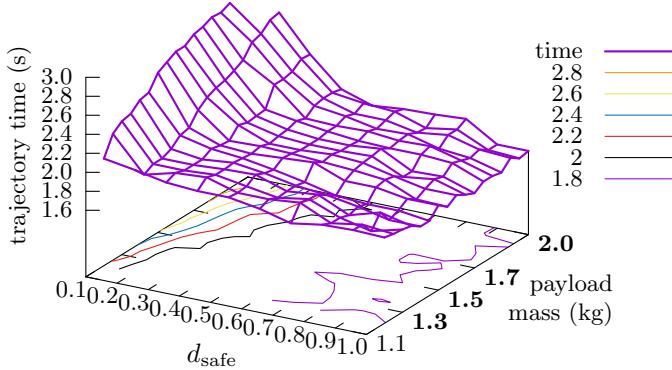


Fig. 7: **Computed trajectory times for varying  $d_{\text{safe}}$  and payload mass.** We compute trajectories for a grid of  $d_{\text{safe}}$  and masses shown on the lower axes, and plot the trajectory times on the vertical axis. With lower  $d_{\text{safe}}$  or higher mass, the learned suction failure constraint causes the optimizer to generate slower motions. The masses used for training (1.3, 1.5, 1.7, and 2.0 kg) are in bold. The plot suggests that the network generalizes to unseen masses.

GOMP-ST with  $h = 6$  and  $d = 0.05$  reliably achieves a 100 % success rate, while the baseline method suction grasps fail in nearly all cases. It is also 16 % to 58 % faster than multimodal GOMP-FIT. The ablation of  $h = 1$  shows the importance of history in learning the constraint—without it, GOMP-ST consistently computes motions that lead to suction failures. In the ablation of  $d_{\text{safe}}$ , we see that lower thresholds result in increased success rate, but reduced speed, while the increased  $d_{\text{safe}}$  results in faster motions and decreased success rate. In addition, we observe that GOMP-ST’s performance when transporting a mass unseen during training ( $m = 1.6$  kg) is comparable with its performance on the objects used during the data collection.

We also study trajectories computed with the learned constraint by varying the mass and  $d_{\text{safe}}$  and plotting the results in Fig. 7. The plot suggests that the network is interpolating between training masses.

## 6 Conclusion

We propose GOMP-ST, an algorithm using learned motion constraints for fast transport of objects held in suction grasps. By learning the constraint from real data, we avoid explicitly modeling difficult-to-model properties such as deformation of the suction cup. We also benefit from the learned model neural-network implementation, as it facilitates automatic generation of gradients needed to linearize the constraint for the solver. Experiments on a physical UR5 suggest that the learned constraint can allow the solver to speed up motions by up to 58 %.

In future work, we will experiment with more complex environments, and include additional inputs to the learned constraint to allow it to adapt to different properties of the grasped object, for example, coefficient of friction and center of mass. We also aim to expand the analytic model to no longer be dependent on quasi-static equilibrium or rigid bodies, and therefore include spring state to the state of the trajectory optimization. While there are compelling reasons to move away from single-suction-cup grippers,

they present additional suction failure modes, such as swinging and torquing out of the gripper. Addressing these failure modes may require additional approaches, such as integration with sampling-based optimization methods. Finally, results in modifying the failure threshold suggest that one could make a trade-off between speed and reliability, but how to beneficially make that trade-off is an open issue.

## Acknowledgments

This research was performed at the AUTOLAB at UC Berkeley in affiliation with the Berkeley AI Research (BAIR) Lab, Berkeley Deep Drive (BDD), the Real-Time Intelligent Secure Execution (RISE) Lab, and the CITRIS “People and Robots” (CPAR) Initiative. We thank our colleagues for their helpful feedback and suggestions. We thank Tae Myung Huh and Michael Danielczuk for their invaluable advice. We thank Adam Lau for his professional photography. We thank Adam Rashid for his help running the physical robot. This article solely reflects the opinions and conclusions of its authors and do not reflect the views of the sponsors or their associated entities.

## Bibliography

- [1] Acharya, P., Nguyen, K.D., La, H.M., Liu, D., Chen, I.M.: Nonprehensile manipulation: a trajectory-planning perspective. *IEEE/ASME Transactions on Mechatronics* **26**(1), 527–538 (2020)
- [2] Bartolini, A., Lombardi, M., Milano, M., Benini, L.: Neuron constraints to model complex real-world problems. In: International Conference on Principles and Practice of Constraint Programming. pp. 115–129. Springer (2011)
- [3] Bernheisel, J.D., Lynch, K.M.: Stable transport of assemblies: Pushing stacked parts. *IEEE Transactions on Automation Science and Engineering* **1**(2), 163–168 (2004)
- [4] Berscheid, L., Kröger, T.: Jerk-limited real-time trajectory generation with arbitrary target states. *arXiv preprint arXiv:2105.04830* (2021)
- [5] Bradbury, J., Frostig, R., Hawkins, P., Johnson, M.J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., et al.: JAX: Autograd and XLA. *Astrophysics Source Code Library* pp. ascl–2111 (2021)
- [6] De Raedt, L., Passerini, A., Teso, S.: Learning constraints from examples. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018)
- [7] Fajemisin, A., Maragno, D., Hertog, D.d.: Optimization with constraint learning: A framework and survey. *arXiv preprint arXiv:2110.02121* (2021)
- [8] Ha, H., Song, S.: FlingBot: The unreasonable effectiveness of dynamic manipulation for cloth unfolding. In: Conference on Robotic Learning (CoRL) (2021)
- [9] Hauser, K.: Fast interpolation and time-optimization with contact. *The International Journal of Robotics Research* **33**(9), 1231–1250 (2014)

- [10] Huh, T.M., Sanders, K., Danielczuk, M., Li, M., Goldberg, K., Stuart, H.S.: A multi-chamber smart suction cup for adaptive gripping and haptic exploration. arXiv preprint arXiv:2105.02345 (2021)
- [11] Ichnowski, J., Avigal, Y., Liu, Y., Goldberg, K.: GOMP-FIT: Grasp-optimized motion planning for fast inertial transport. In: 2022 International Conference on Robotics and Automation (ICRA). IEEE (2022), (*to appear*)
- [12] Ichnowski, J., Avigal, Y., Satish, V., Goldberg, K.: Deep learning can accelerate grasp-optimized motion planning. *Science Robotics* **5**(48) (2020)
- [13] Ichnowski, J., Danielczuk, M., Xu, J., Satish, V., Goldberg, K.: GOMP: Grasp-optimized motion planning for bin picking. In: 2020 International Conference on Robotics and Automation (ICRA). IEEE (May 2020)
- [14] Kalakrishnan, M., Chitta, S., Theodorou, E., Pastor, P., Schaal, S.: STOMP: Stochastic trajectory optimization for motion planning. In: 2011 IEEE international conference on robotics and automation. pp. 4569–4574. IEEE (2011)
- [15] Kavraki, L.E., Svestka, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high dimensional configuration spaces. *IEEE Trans. Robotics and Automation* **12**(4), 566–580 (1996)
- [16] Kolluru, R., Valavanis, K., Hebert, T.: Modeling, analysis, and performance evaluation of a robotic gripper system for limp material handling. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **28**(3), 480–486 (1998)
- [17] Kudł, P., Pawlak, T.P.: One-class synthesis of constraints for mixed-integer linear programming with c4. 5 decision trees. *Applied Soft Computing* **68**, 1–12 (2018)
- [18] Kuntz, A., Bowen, C., Alterovitz, R.: Fast anytime motion planning in point clouds by interleaving sampling and interior point optimization. *Robotics Research* pp. 929–945 (2020)
- [19] LaValle, S.M., Kuffner, J.J.: Rapidly-exploring random trees: Progress and prospects. In: Donald, B.R., Others (eds.) *Algorithmic and Computational Robotics: New Directions*, pp. 293–308. AK Peters, Natick, MA (2001)
- [20] Lim, V., Huang, H., Chen, L.Y., Wang, J., Ichnowski, J., Seita, D., Laskey, M., Goldberg, K.: Planar robot casting with Real2Sim2Real self-supervised learning. arXiv preprint arXiv:2111.04814 (2021)
- [21] Lombardi, M., Milano, M., Bartolini, A.: Empirical decision model learning. *Artificial Intelligence* **244**, 343–367 (2017)
- [22] Luh, J.Y., Walker, M.W., Paul, R.P.: On-line computational scheme for mechanical manipulators (1980)
- [23] Luo, J., Hauser, K.: Robust trajectory optimization under frictional contact with iterative learning. *Autonomous Robots* **41**(6), 1447–1461 (2017)
- [24] Lynch, K.M., Mason, M.T.: Dynamic underactuated nonprehensile manipulation. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS'96. vol. 2, pp. 889–896. IEEE (1996)
- [25] Lynch, K.M., Mason, M.T.: Dynamic nonprehensile manipulation: Controllability, planning, and experiments. *The International Journal of Robotics Research* **18**(1), 64–92 (1999)
- [26] Mahler, J., Matl, M., Liu, X., Li, A., Gealy, D., Goldberg, K.: Dex-Net 3.0: Computing robust vacuum suction grasp targets in point clouds using a new analytic model and deep learning. In: 2018 IEEE International Conference on robotics and automation (ICRA). pp. 5620–5627. IEEE (2018)

- [27] Maragno, D., Wiberg, H., Bertsimas, D., Birbil, S.I., Hertog, D.d., Fajemisin, A.: Mixed-integer optimization with constraint learning. arXiv preprint arXiv:2111.04469 (2021)
- [28] Mucchiani, C., Yim, M.: Dynamic grasping for object picking using passive zero-dof end-effectors. IEEE Robotics and Automation Letters **6**(2), 3089–3096 (2021)
- [29] Park, C., Pan, J., Manocha, D.: ITOMP: Incremental trajectory optimization for real-time replanning in dynamic environments. In: Twenty-Second International Conference on Automated Planning and Scheduling (2012)
- [30] Pham, H., Pham, Q.C.: Critically fast pick-and-place with suction cups. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 3045–3051. IEEE (2019)
- [31] Pham, Q.C., Caron, S., Nakamura, Y.: Kinodynamic planning in the configuration space via admissible velocity propagation. In: Robotics: Science and Systems. vol. 32 (2013)
- [32] Ratliff, N., Zucker, M., Bagnell, J.A., Srinivasa, S.: CHOMP: Gradient optimization techniques for efficient motion planning. In: 2009 IEEE International Conference on Robotics and Automation. pp. 489–494. IEEE (2009)
- [33] Ruggiero, F., Lippiello, V., Siciliano, B.: Nonprehensile dynamic manipulation: A survey. IEEE Robotics and Automation Letters **3**(3), 1711–1718 (2018)
- [34] Schulman, J., Ho, J., Lee, A.X., Awwal, I., Bradlow, H., Abbeel, P.: Finding locally optimal, collision-free trajectories with sequential convex optimization. In: Robotics: Science and Systems. pp. 1–10 (2013)
- [35] Srinivasa, S.S., Erdmann, M.A., Mason, M.T.: Using projected dynamics to plan dynamic contact manipulation. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems. pp. 3618–3623. IEEE (2005)
- [36] Stuart, H.S., Bagheri, M., Wang, S., Barnard, H., Sheng, A.L., Jenkins, M., Cutkosky, M.R.: Suction helps in a pinch: Improving underwater manipulation with gentle suction flow. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) pp. 2279–2284 (2015)
- [37] Toussaint, M.: Newton methods for k-order markov constrained motion problems. arXiv preprint arXiv:1407.0414 (2014)
- [38] Valencia, A.J., Idrovo, R.M., Sappa, A.D., Guingla, D.P., Ochoa, D.: A 3d vision based approach for optimal grasp of vacuum grippers. In: 2017 IEEE International Workshop of Electronics, Control, Measurement, Signals and their Application to Mechatronics (ECMSM). pp. 1–6 (2017)
- [39] Wang, C., Wang, S., Romero, B., Veiga, F., Adelson, E.: SwingBot: Learning Physical Features from In-hand Tactile Exploration for Dynamic Swing-up Manipulation. In: Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS) (2020)
- [40] Zeng, A., Song, S., Lee, J., Rodriguez, A., Funkhouser, T.: TossingBot: Learning to throw arbitrary objects with residual physics. IEEE Transactions on Robotics **36**(4), 1307–1319 (2020)
- [41] Zhang, H., Ichnowski, J., Seita, D., Wang, J., Goldberg, K.: Robots of the lost arc: Learning to dynamically manipulate fixed-endpoint ropes and cables. In: Proc. IEEE Int. Conf. Robotics and Automation (ICRA) (2021)