Effector Shape and Motion Optimization

by

Rebecca H. Jiang

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

© Rebecca Jiang. All rights reserved.
The author hereby grants to MIT and The Charles Stark Draper Laboratory, Inc. permission to reproduce and to distribute publicly paper and electronics copies of this thesis document in whole or in part in any medium now known or hereafter created.

Author . . . .
Department of Aeronautics and Astronautics
May 17, 2022

Certified by . . . .
Alberto Rodriguez
Associate Professor at MIT
Thesis Supervisor

Certified by . . . .
Ravi Gondhalekar
Senior Member of the Technical Staff at Draper
Thesis Supervisor

Accepted by . . . .
Jonathan P. How
R. C. Maclaurin Professor of Aeronautics and Astronautics
Chair, Graduate Program Committee

Effector Shape and Motion Optimization

by

Rebecca H. Jiang

Submitted to the Department of Aeronautics and Astronautics
on May 17, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

In this thesis, methods are proposed for co-optimizing the shape and motion of robotic effectors for planar tasks. An effector is a device, typically at the end of a robotic arm, used to interact with the environment. While planning object and robot-object contact trajectories is extensively studied, designing an effector that can execute the planned trajectories receives less attention. As such, this thesis includes a framework that synthesizes an object trajectory and object-effector contact trajectory into an effector trajectory and shape that (a) does not penetrate the object, (b) makes contact with the object as specified, and (c) optimizes a user-specified objective. This simplifies manipulator control by encoding task-specific contact information in the effector's geometry. The key insight is posing these requirements as constraints in the effector's reference frame, preventing the need for explicit parameterization of the effector shape. This prevents artificial restrictions on the shape design space. Importantly, it also facilitates posing the shape and motion design problem as a tractable nonlinear program. This method is particularly useful for problems where the shape of the effector surface must be precisely chosen to achieve a task. This work is then extended to parallel-jaw grasping problems, in which grasp stability is considered while optimizing over contact locations, effector shape, and grasp configuration. This provides a path forward for future work in which effectors with multiple internal degrees of freedom are co-optimized with motion. Methods are demonstrated on example problems, including jar-opening, picking up objects in constrained spaces, and stably grasping sets of nonconvex objects. The algorithms' results and computational cost are evaluated. A physical experiment demonstrates a robotic arm picking up a screwdriver from a table using a tool that was designed using the proposed framework and manufactured to the derived shape.

Thesis Supervisor: Alberto Rodriguez
Title: Associate Professor at MIT

Thesis Supervisor: Ravi Gondhalekar
Title: Senior Member of the Technical Staff at Draper

Acknowledgments

I thank Draper, and the Draper Scholar Program, for funding this work. Draper has supported me in exploring research directions at the intersection of robotics and space applications, while learning from a network of seasoned engineers.

Thanks to Ravi Gondhalekar, my advisor at Draper, for being deeply supportive. Your technical knowledge and creative ideas have enabled our research and inspired me to think critically in these last two years. You have been patient, spending many cumulative hours talking with me, discussing my good ideas, discussing my bad ideas, helping me become a good researcher, and engaging in a little intellectual frolic.

I thank Alberto Rodriguez, my advisor at MIT, for helping me acclimate to the world of robotics research. You have continuously exceeded how kind, understanding, and supportive I expect an advisor to be. In the ups and downs of research, knowing that I have you on my team assures me that we will ultimately accomplish our goals.

To Neel Doshi, thank you for your phenomenal mentorship. Our weekly discussions brought many creative ideas and led me to keep educating and challenging myself as we delved into them.

To my academic advisor Richard Linares, thank you for supporting my journey in the AeroAstro department while I research in a Mechanical Engineering lab.

Thank you Andy Ruina, for pushing me to go to grad school, when I had previously not seen myself as a researcher. I know that my life would look very different now had I not had your support and advice in the years before I came to MIT.

John Wirzburger, thank you for having been an incredible leader during my time at The Johns Hopkins University Applied Physics Lab, and for helping me grow and find my direction in career and life. Your passion for space exploration and your engineering ideology stay with me and will continue to influence me throughout my career.

Finally, an acknowledgement to my cats, Emmy and Fritz, who have made many suggested edits to my code and writing that I ultimately undid, and who have been incredible officemates during the work-from-home days.

Contents

List of Figures

List of Tables

Nomenclature

N_t	Number of timesteps
N_c	Number of contacts (Chap. 2)
$N_c[k]$	Number of contacts at index k (Chap. 3)
N_d	Number of allowed effector degrees of freedom
N_p	Number of parameters used to define effector state
N_{iter}	Number of re-initializations of the NLP solver
T_s	Time taken to solve a problem
E	Effector frame (Chap. 2)
E_j	Effector frame for jth jaw (Chap. 3)
O	Object frame
W	World frame
$\Psi^A[k]$	Object in reference frame A at index k
$\mathbf{d}_i^A[k]$	Tangent for contact i in reference frame A at index k
$\mathbf{p}_i^A[k]$	Position of contact i in reference frame A at index k
$\mathbf{p}_E^W[k]$	Position of the effector in the world frame at index k
$s_i[k]$	Path length parameter for contact i at index k
$\Omega^A[j]$	The jth obstacle, in the reference frame A
$\theta_E[k]$	Orientation of the effector at index k
$\mathcal{P}$	Concatenation of effector positions $p_E^W[k]$ over all k, $\mathcal{P} := \{\mathbf{p}_E^W[0], ..., \mathbf{p}_E^W[N_t]\}$
Θ	Concatenation of effector orientations $\theta_E[k]$ over all k, $\Theta := \{\theta_E[0], ..., \theta_E[N_t]\}$
$\mathcal{S}$	Concatenation of path length parameters $s_i[k]$ over all i and k, $\mathcal{S} := \{s_0[1], ..., s_0[N_t], ..., s_{N_c-1}[1], ..., s_{N_c-1}[N_t]\}$
$\phi[k]$	Signed distance function of the composite shape at index k
$h[k]$	Timestep at index k
$\dot{\mathbf{p}}_{max}$	Limit on rate of change of effector position

$\dot{\theta}_{max}$	Limit on rate of change of effector orientation
J_x	A user-specified cost function, distinguished by some name placeheld by x
J_G	Grasp quality function
J_Px	A user-defined cost function for a parallel-jaw effector, distinguished by some name placeheld by x
w	A scalar to scale the relative weights of cost components
$L[k]$	A characteristic length for object of index k
$v_i[k]$	Object edge assignment for contact i at index k
$M_j[k]$	A set of indices for contacts belonging to jaw j on object k
$\gamma[k]$	Jaw opening distance at index k
Γ	Concatenation of jaw opening distances $\gamma[k]$ over all k, $\Gamma := \{\gamma[0], ..., \gamma[N_t]\}$
$t_i[k]$	the ith contact's position coordinate
$\mathbf{t}[k]$	Concatenation of contact position coordinates $t_i[k]$ over all contacts i at index k, $\mathbf{t}[k] := [t_0[k], ..., t_{N_c[k]-1}[k]]^T$
$\mathcal{T}$	Concatenation of all contact position coordinates $\mathbf{t}[k]$ over all k, $\mathcal{T} := \{\mathbf{t}[0], ..., \mathbf{t}[N_t]\}$
$M_j[k]$	Set of contact indices belonging to jaw j at index k
$\tilde{\Psi}^E[k]$	The swept effector-frame object at index k
$\mathbf{r}$	Rigid-body object displacements
$\mathbf{q}$	Jaw displacements
$c_{i,n}$	Force along the contact normal at contact i
$c_{i,t}$	Force along the contact tangent at contact i
$\mathbf{c}$	Concatenation of contact forces
$d_{i,n}$	Spring displacement along the contact normal at contact i
$d_{i,t}$	Spring displacement along the contact tangent at contact i
$\mathbf{d}$	Concatenation of spring displacements
$\mathcal{G}$	Grasp matrix
$\mathcal{J}$	Hand matrix
F	Preload force applied to jaws
μ	Coefficient of friction

$\mathbf{w}$	External wrench applied to object
B	A matrix that selects alternate rows of another matrix when pre-multiplying it

List of Acronyms

DARPA	Defense Advanced Research Projects Agency
DLR	German Aerospace Center
DOF	degree of freedom
KKT	Karush–Kuhn–Tucker
MINLP	mixed-integer nonlinear program
NASA	National Aeronautics and Space Administration
NLP	nonlinear program
POD	Payload Orbital Delivery
PA	Persistent Asset
QP	quadratic program
SDF	signed distance function
TA	Technology Area

Chapter 1

Introduction

1.1 Motivation

A growing number of in-space technology goals, including large telescopes, artificial gravity, and solar-electric propulsion, involve assembling large structures in space [8]. However, launch vehicle size and weight limits inhibit launching large structures. Furthermore, launches are expensive, and crewed servicing missions are rare; until robots are capable of a broad range of complex spacecraft servicing tasks, there is little opportunity to repair or improve in-space assets. Consequently, currently: (a) engineering conservatism raises cost while sacrificing performance, (b) one failure can cause loss of mission, (c) flying technology is not updated to match new developments, and (d) fuel limitations restrict capability and lifetime. In anticipation of future robotic capabilities, the National Aeronautics and Space Administration (NASA) has proposed a new paradigm where spacecraft known as Persistent Assets (PAs) [14] are intended to have long lifetimes, be assembled in space, be upgraded as technology improves or as budget becomes available, and be serviced and maintained as needed. In a complementary concept by the Defense Advanced Research Projects Agency (DARPA), Payload Orbital Deliveries (PODs) are sent to a servicing spacecraft on orbit, containing spare tools or parts, new components to augment or upgrade a client spacecraft, or modular building blocks for assembling a new spacecraft [49].

Robotic effectors will hold a central role in assembling and servicing PAs. No single effector can perform all necessary tasks. For example, DARPA's Robotic Servicing of Geosynchronous Satellites mission includes the German Aerospace Center (DLR) Spacehand, as well as separate tools for POD capture, grappling a client spacecraft, and applying controlled forces and torques [1]. Likewise, NASA's On-orbit Servicing, Assembly, and Manufacturing 1 mission includes individual tools for grasping and berthing, and for manipulating and cutting thermal blankets [50]. Tailoring effectors to tasks raises the key question: **What is the optimal effector for a given task?** Indeed, the NASA Technology Roadmap, Technology Area (TA) 4.3.2 – Robotic Manipulation – calls for design of physical manipulators *and* their motions [5]. Obviously, designing an effector involves selecting its physical form. However, its intended motion is an equally significant part of the solution. The form of a screwdriver is meaningless without the intention to hold the tip to the screw head and rotate, as is the shape of a parallel-jaw gripper surface without the intention to position the jaws around an object and close. In this thesis, methods are proposed for finding the optimal manipulator form and motion for a given task.

TA 4.3.2 also emphasizes the need for grasping various object geometries. The proposed method addresses precisely this problem, with a tool that synthesizes optimized effectors with stable grasps for a collection of input objects. This framework can help minimize the number of distinct tools needed for a set of objects, a priority mentioned in TA 4.3.2. Indeed, grasping is central in recent demonstrations. Orbital ATK's Commercial Infrastructure for Robotic Assembly and Services demonstration used a parallel-jaw "Universal Grabber Tool" to manipulate trusses and solar arrays [2]. Langley Research Center's SAMURAI robot grasps a truss strut with two grippers simultaneously [15]. SpiderFab (by Tethers Unlimited) used a gripper with the particular form and compliance needed to align a truss strut, allowing for error in end effector position [27]. DLR presents gripper fingers shaped to create form-closure grasps on assembly kit parts and a screwdriver tool [42]. All of these effectors, manually designed, must precisely and stably align assembly elements. The framework developed in this thesis provides a preliminary step in the direction of optimizing these effectors.

1.2 Design and use of effectors

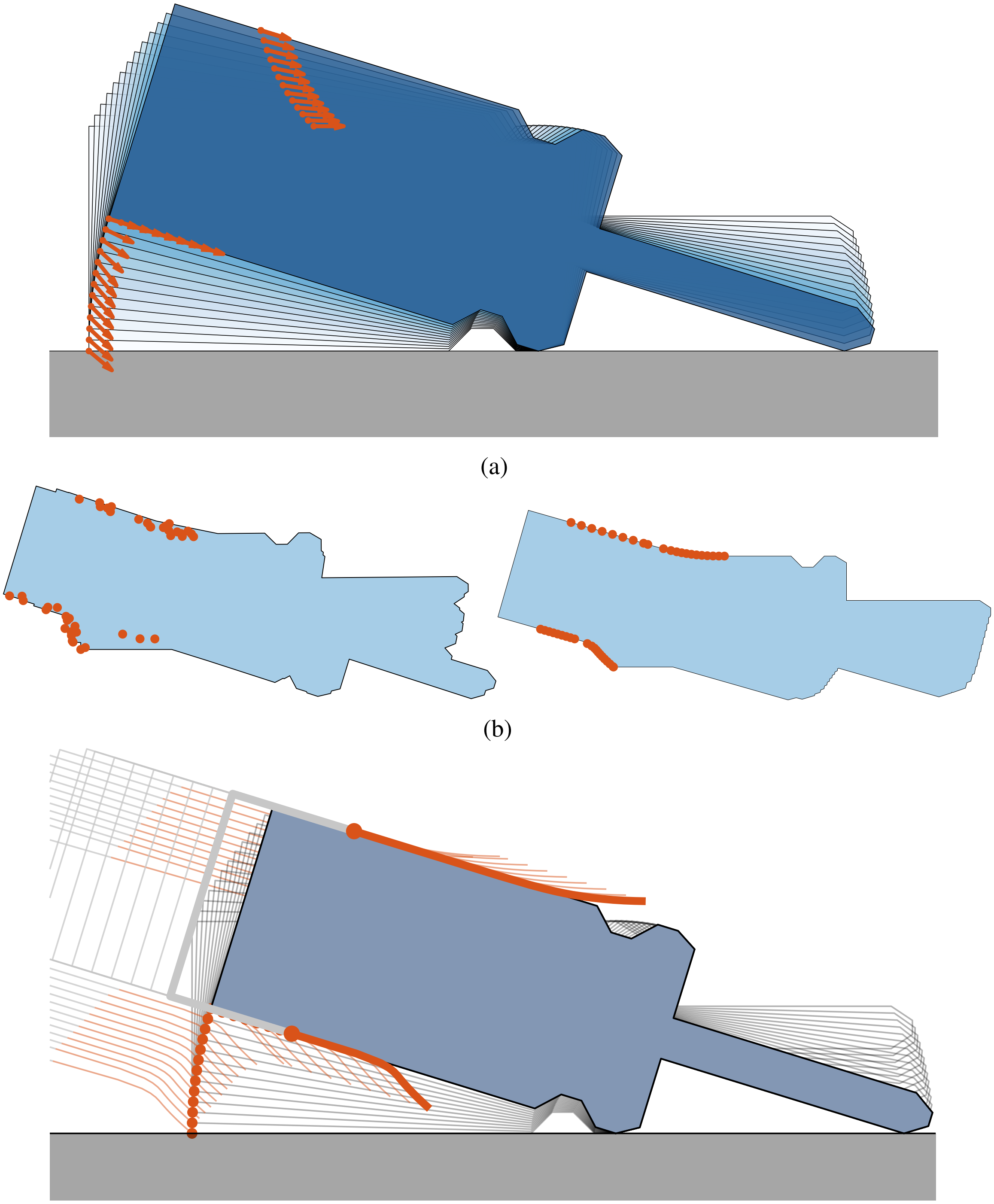
The performance of a robot is a matter of both its morphology and control: The optimal effector lies in the shared design space of shape and motion. However, in robotics we typically select an effector without optimizing for an intended task, and only subsequently design a corresponding effector motion [37, 38, 41, 52, 56]. Pre-selecting an effector preemptively constrains the space of feasible motions unnecessarily.

Several works avoid this restriction by taking an object-centric approach, optimizing directly over contact locations and object trajectory [3, 12, 16, 33, 35]. However, these frameworks result in disembodied contact points; the shape and motion of the effector that contacts these points remain unspecified. One solution is to use a dexterous hand with an individual finger to track each contact point [12, 35]; however, this incurs substantial mechanical and control complexity. Instead, particularly in settings where the same task is to be repeated many times, a minimally actuated task-specialized effector can reduce complexity and simplify task execution.

In this thesis, a framework is presented for co-optimizing shape and motion of this task-specialized effector given trajectories for the object and object-effector contact points. Co-optimizing effector shape and motion is challenging. This is largely due to difficulties in representing the effector shape in a way that (a) can be integrated into an optimization algorithm, (b) is expressive enough to allow the emergence of useful design features, (c) is conducive to expressing constraints, and (d) is computationally tractable. The proposed framework addresses these issues by circumventing the need to explicitly parameterize the effector shape.

This is accomplished by leveraging a key insight from Rodriguez and Mason [43, 44], who examine how task-specific contact trajectories induce a relationship between the effector's motion and the shape of its contact surface. The effector contact surface must meet prescribed points in space while traveling along its trajectory. Equivalently, these points, when represented in a reference frame fixed to the effector, must lie on its contact surface. This allows the contact points that encode the desired task to also implicitly represent the effector's shape as a function of its motion.

The key contribution of this thesis is to use this insight to pose the simultaneous effector shape and motion optimization as a tractable nonlinear program (NLP) over only motion variables. Focusing first on rigid effectors with given contact trajectories, this thesis introduces two key constraints on these motion variables that enforce feasibility of the physical effector: *continuity* and *non-penetration*. Continuity enforces a viable effector shape, and non-penetration globally prevents object-effector and effector-environment interpenetration. Fig. 1-1 summarizes the proposed framework for rigid effectors. The inputs to the framework are shown in Fig. 1-1a. In an example infeasible candidate solution, visualized in the effector frame in Fig. 1-1b (left), some contact points penetrate the object, and contact points are scattered and not aligned. These issues are corrected through the optimization process, leading to Fig. 1-1b (right). Fig. 1-1c shows the output, an optimized feasible shape and motion of the effector. The cost function is specified by the user, allowing the framework to flexibly synthesize effectors with application-specific desirable qualities. This thesis also includes an extension of the framework for optimizing parallel-jaw effectors, establishing a path forward to incorporate internal effector degrees of freedom (DOFs), force balance, and optimization over contact location.



(a)

(b)

(c)

Figure 1-1: a) Input: Desired object and contact constraints in world frame. b) Infeasible (left) and feasible (right) candidate solutions during the optimization process, visualized in the effector frame. Blue: union of object representations in the effector frame across timesteps. c) Final effector solution.

1.3 Related work

1.3.1 Effector shape and motion optimization

A common method of co-optimizing shape and motion is to use an existing geometry and optimize a few tunable parameters, as is commonly done in legged locomotion [13, 18, 21, 34, 39, 40, 46, 54]. This approach has been demonstrated for manipulation tasks using gradient-based optimization [26], sampling-based motion planning [7], the implicit function theorem [23], and reinforcement learning [11]. While these works have achieved success in their intended applications, their physical representations are limited; they are not expressive enough to enable discovering geometries vastly different from the initial design, or permit characterizing fine details of a contact surface whose precise form determines its feasibility.

There is less work on shape and motion optimization using more expressive geometric representations. Like in the proposed approach, Xu et al. [57] use gradient-based optimization to co-design effector shape and motion. They begin with an initial geometry made up of effector components from a database, and distort these components using a cage-based deformation technique [28]. Using their own dynamics simulator that is differentiable with respect to the cage vertices, they simultaneously optimize over morphology and control. Taylor and Rodriguez [51] formulate an NLP to optimize spline parameters representing the shape and motion of a rigid planar effector performing dynamic manipulation tasks. However, they do not handle multiple contacts, do not permit sliding contact, and do not enforce global non-penetration, all of which would be additions that significantly increase problem size and introduce nonconvexities. Moreover, their approach requires significant manual supervision of spline parameters and regularization constraints. While the approach proposed in this thesis does not enforce dynamics, as they are assumed to be satisfied in the input trajectories, the presented approach allows multiple contacts and enforcement of global non-penetration, and can extend to effectors with internal DOFs as discussed in Chap. 3.

The representations in Xu et al. [57] and Taylor and Rodriguez [51] are still more restrictive than the proposed method, which does not explicitly parameterize the effector shape. However, Kawaharazuka et al. [29] control effector shape pixel-by-pixel, providing a very general shape representation. They use a neural network to optimize the shape and motion of a rigid planar tool represented by an image. Their method succeeds for the simple task of sliding a circular object from one point on a surface to another, but requires collecting data from physical experiments requiring “randomized” tools to be manufactured. Additionally, their work does not explore more complex tasks.

1.3.2 Parallel-jaw gripper optimization

Ha et al. [22] and Alet et al. [4] present learning-based frameworks for optimizing parallel-jaw grippers. Both of these works evaluate grasps by simulation of the grasping process, encouraging robustness by simulating with multiple initial object poses. Like in the approach proposed in this thesis, Alet et al. design grippers for sets of objects. These two works show good tolerance to uncertainty, but do not fully leverage model information or the powerful tools of grasp stability analysis to design grippers with highly tailored geometry. Schwartz et al. [47] likewise define a grasp metric based on success of grasping simulations (with a single object), within an optimization loop in a model-based approach. They avoid being constrained by effector geometry parameterization by using the negative of the object shape to generate the matching effector geometry, similarly to the method proposed in Sec. 3.1.2. Yako et al. [58] and Wolniakowski et al. [55] select dimensions and other mechanical parameters for effectors for grasping. Wolniakowski et al. simulate grasps to evaluate their quality, while Yako et al. use a potential energy map approach to understand grasping behavior without simulation. These low-dimensional parameterizations are successful at selecting some effector characteristics, but are limited in their expressivity when designing effectors without any priors. Finally, Brown et al. [10] design grippers by arranging pins to create form closure grasps, rejecting solutions that cannot geometrically close the jaws around the object. Similarly to the present work, they reason about grasps from a point-contact perspective, geometrically reason about non-penetration during jaw closure, and score grasp quality. However, they assume a given object orientation within the jaws, and thus lack this important optimization space freedom.

1.4 Outline of this thesis

This thesis first describes a framework for optimizing rigid effectors for contact trajectory satisfaction in Chap. 2, and subsequently in Chap. 3 describes extensions of the concepts from this framework to allow internal effector DOFs, consideration of force balance, and optimizing over contact locations in the context of grasping. Sec. 2.1 develops the core concept of this work, posing the shape and motion problem as an NLP over only motion variables. The NLP constraints are developed in Secs. 2.2 and 2.3, and the NLP itself for rigid effectors is composed in Sec. 2.4. Sec. 2.5 gives example results, and Sec. 2.6 shows a real-world experiment. Computation is discussed in Sec. 2.7.

In figures throughout Chap. 2, an object color gradient from white to blue signifies progression in time, orange points represent object-effector contact points, orange arrows represent contact tangents, orange curves represent effector contact surfaces, and gray curves are (non-optimized) connectors for clarity. Gray regions are user-defined obstacles.

Sec. 3.1 explains the extension of the framework from Chap. 2 to grasping, adding the notion of grasp stability in Sec. 3.2, updating the non-penetration constraint in Sec. 3.3, and formulating the updated NLP in Sec. 3.4. Sec. 3.5 gives example results for the grasping framework. Computation is discussed in Sec. 3.6

In figures throughout Chap. 3, red features refer to the left jaw, and orange features refer to the right jaw. Colored points represent contact points. An object color gradient from white to blue signifies progression in time. The effector jaw axis is horizontal. Unless otherwise specified the effectors shown are generated using the approach outlined in Sec. 3.1.2.

Chapter 2

Contact trajectory satisfaction

2.1 The shape and motion problem

This section discusses the proposed framework, including the problem formulation and representation, and how feasibility of a candidate solution is assessed. The *non-penetration* and *continuity* constraints are introduced in the context of an example problem, the design of a rock-climbing cam.

2.1.1 Problem formulation and notation

The following items are required problem inputs:

- A polygonal object, $\Psi^O[k]$, in its own reference frame (superscript O), at all times $k = 0, ..., N_t$. The object geometry may evolve over time.

- A discretized trajectory of the object: the orientations $\theta_O[k]$ and positions $\mathbf{p}_O^W[k]$ of the object (subscript O) in the world frame (W) at all times.

- Contact constraints: contact point position, $\mathbf{p}_i^O[k]$, and contact tangent, $\mathbf{d}_i^O[k]$ at all times, and for all contacts $i = 0, ..., N_c - 1$, where N_c is the number of contacts. The contact tangent is colinear with the object edge unless the contact point is an object vertex. In this case, the tangent and its antiparallel direction must not point into the object, but are otherwise free. The effector must contact the object at the contact points with its surface tangent to the contact tangents.

- A cost function on the effector motion, shape, or function. Sec. 2.5 shows examples.
- The allowed effector DOFs. The effector is rigid and has between one and three DOFs (N_d) in the plane.

If a solution is found, the framework outputs:

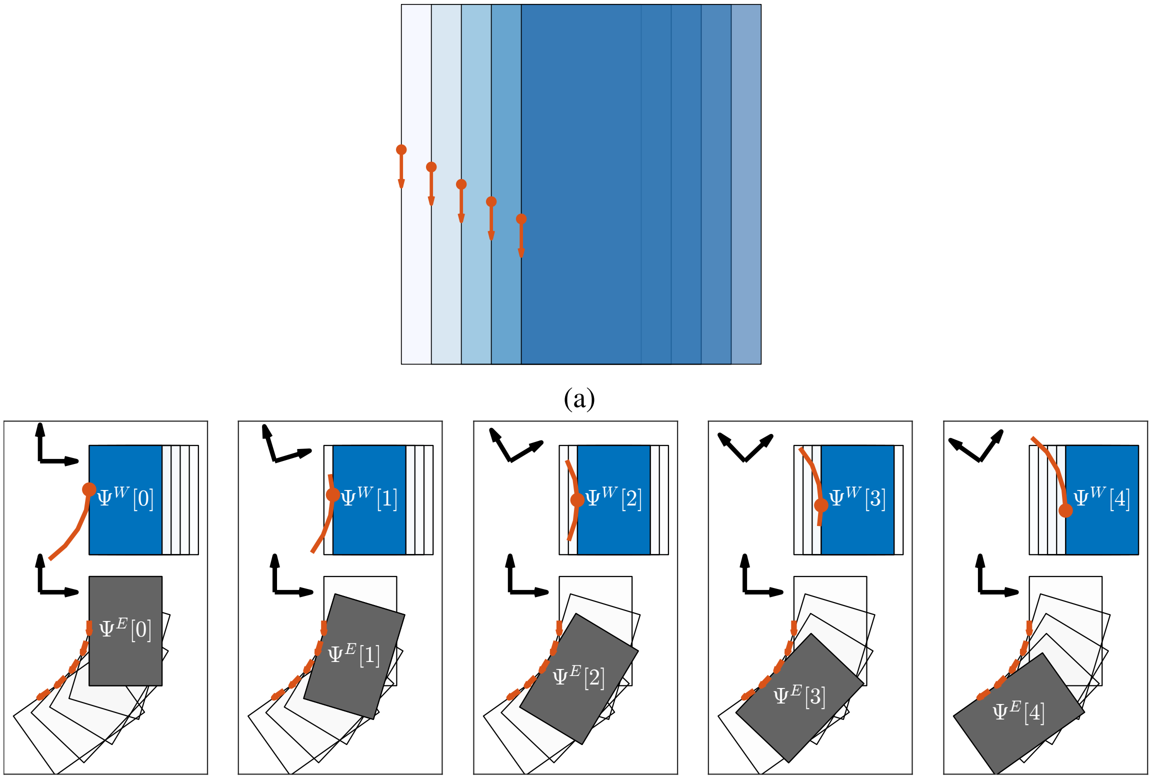
- $\mathbf{p}_E^W[k]$ and $\theta_E[k]$, the position and orientation of the effector (E), at all times. These trajectories are abbreviated as $\mathcal{P} := \{\mathbf{p}_E^W[0], ..., \mathbf{p}_E^W[N_t]\}$, $\Theta := \{\theta_E[0], ..., \theta_E[N_t]\}$.
- The value, for all timesteps, and for all contacts, of a path length parameter, $s_i[k]$, that modulates the distances between contact points along the contact surface, motivated in Sec. 2.2. $\mathcal{S} := \{s_0[1], ..., s_0[N_t], ..., s_{N_c-1}[1], ..., s_{N_c-1}[N_t]\}$.
- N_c sets of points, each discretizing an effector contact surface that meets a contact constraint at all times when traveling along the output effector trajectory.

The proposed approach, detailed in Sec. 2.1.2, relies on expressing variables in the *effector frame* (E). Let $\Psi^E[k]$, $\mathbf{p}_i^E[k]$, and $\mathbf{d}_i^E[k]$ denote the object, position of the ith contact point, and ith contact tangent, represented in E at time k.

2.1.2 Approach

For a candidate effector motion, the object and the contact constraints are represented in a frame fixed to the effector. As is discussed in this section, this prevents the need to parameterize the effector shape, circumventing some key limitations in related works. Instead, in the effector frame E, two core constraints are formulated, the *non-penetration* and *continuity* constraints, that ensure that the effector motion leads to a feasible effector shape.

In the continuous problem ($N_t \to \infty$), the continuum of contact points, when represented in the effector frame, must coincide with the effector surface in order for the effector to meet these contact points. Thus, we can treat the discrete points $\mathbf{p}_i^E[0], ..., \mathbf{p}_i^E[N_t]$ as a discretization of the effector surface corresponding to an effector motion. We optimize over



(a)

(b)

Figure 2-1: a) Object and contact constraints for all times for the rock-climbing cam problem, in the world frame. The goal is to design a surface that, when articulated, meets all five contacts shown (see Fig. 16 in [43] for further explanation). b) Illustration of a solution for the effector shape and motion in the rock-climbing cam problem. Blue object snapshots are shown in the world frame. The contact surface of the effector is shown, but the rest of the effector body is not defined; its motion (pure rotation) is demonstrated by the rotation of effector-frame basis vectors. Gray object snapshots with their corresponding contact constraints are shown in the effector frame. Effector-frame basis vectors are shown, static in the effector frame. Notice that the contact points and tangents define the contact surface.

effector motion and extract the corresponding effector surface geometry from the contact points.

As an example to demonstrate what makes an effector shape and motion feasible, consider the rock-climbing cam design problem from Fig. 16 of Rodriguez and Mason [43]. A rock-climbing cam is a device with two spring-loaded surfaces that wedge in rock cracks. In order to transmit the necessary forces, the cam is required to contact rock walls at a constant contact angle, independent of the wall distance. The contact angle is defined as the angle between the horizontal and the line that passes through the pivot and the contact point. Fig. 2-1a shows input information for this problem, and Fig. 2-1b visualizes the output in E and in W. We take advantage of the symmetry of the problem and design only one cam. A coarse time discretization is used for the sake of clear visual illustrations.

The continuity constraint. In Fig. 2-1b, $\mathbf{p}_0^E[k]$ form the same shape as the effector solution. This is because the contact points, represented in the effector frame, make up the effector surface geometry. We use the contact points in the effector frame as the discrete representation of the continuous effector surface. In Fig. 2-1b, as previously in Fig. 1-1b (right), $\mathbf{p}_0^E[k]$ align in a curve that respects the contact tangents. In contrast, the contact points in Fig. 1-1b (left) do not, and thus the candidate motion shown there is infeasible. This behavior is constrained by the continuity constraint, formalized in Sec. 2.2.

The non-penetration constraint. In Fig. 2-1b, when represented in the effector frame, the contact points, $\mathbf{p}_0^E[j]$, do not penetrate the object, $\Psi^E[k]$, in this case the gray wall, at any time. This condition must hold for any feasible solution; otherwise, the effector would have to penetrate the object at some time in order to meet a contact constraint at some other time. The non-penetration constraint is formalized in Sec. 2.3.

2.2 Continuity constraints

Sec. 2.1.2 suggests that the effector contact surface can be recovered by connecting consecutive contact points, $\mathbf{p}_i^E[k]$. However, in order to respect the contact tangents while doing so, an additional constraint is needed.

Rodriguez and Mason [43, 44] determine an effector surface geometry by calculating an integral curve in a vector field of contact tangents, relying on the assumption that these contact tangents align in a way that allows this. The framework proposed in this thesis borrows the key insight that the effector surface must be an integral of the continuum of tangents, and imposes this via the contact points. That is, the proposed method enforces that, in the effector frame, the contact points approximate an integral of the contact tangents. This is referred to as the *continuity constraint*, and is enforced via the trapezoidal

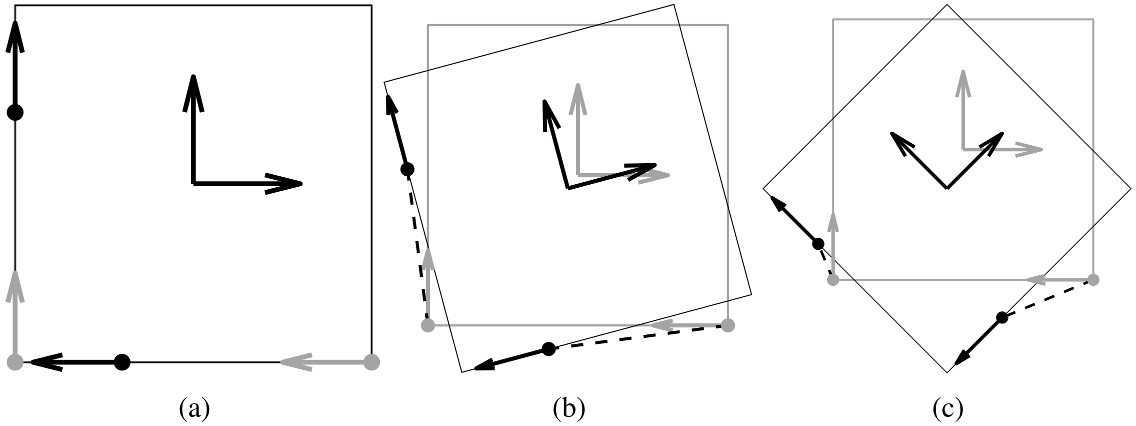


Figure 2-2: Example problem with $N_t = 1, N_c = 2, N_d = 3$. $k = 0$ and $k = 1$ represented in gray and black respectively, with an unrealistically large timestep for the sake of illustration. a) Object and contact constraints shown in the world frame. b-c) Effector-frame representations of continuity-feasible solutions with $\theta_E[0] = 0^o$ and $\theta_E[1] = -15^o$ (b), $\theta_E[1] = -45^o$ (c). Black dashed lines are parallel to the averages of contact tangents from times 0 and 1 (RHS of Eq. (2.1)). The shown solutions are continuity-feasible because indeed these quantities are equal to the displacements of contact points between times 0 and 1 (LHS of Eq. (2.1)).

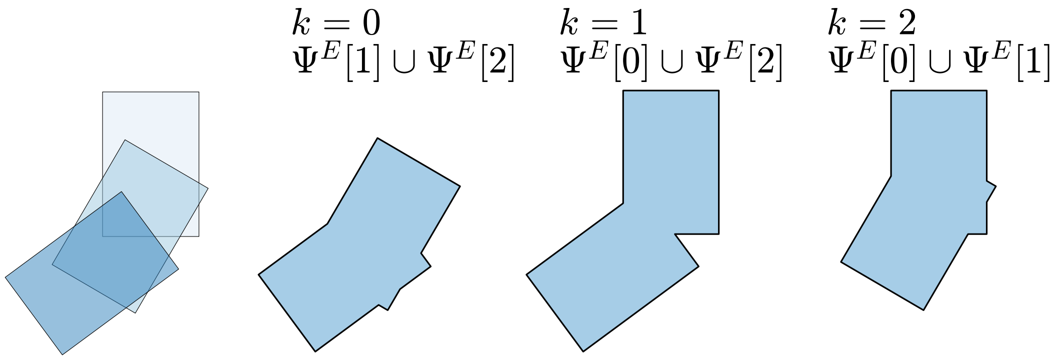
rule for numerical integration [6]:

$$\mathbf{p}_i^E[k] - \mathbf{p}_i^E[k-1] = \frac{s_i[k]}{2}\left(\mathbf{d}_i^E[k-1] + \mathbf{d}_i^E[k]\right). \tag{2.1}$$

Here, $s_i[k]$ is a free parameter that determines how far the ith contact point moves in the effector frame between times $k-1$ and k. This constraint is similar to the trapezoidal collocation scheme [9], frequently used in trajectory optimization. This continuity constraint formulation creates $N_t \cdot N_c$ optimization variables, $s_i[k]$. Fig. 2-2 shows a toy problem to help visualize the continuity constraint. In this problem, where $N_d - N_c = 1$, specifying $\theta_E[1]$ fully determines the effector state at time $k = 1$ relative to the state at $k = 0$.

2.3 Non-penetration constraints

As discussed in Sec. 2.1.2, in order for the candidate effector shape and motion to not cause effector penetration of the object, each contact point must lie outside of the object in the effector frame at each time. The signed distance function (SDF) is used to express this


$k = 0$
$\Psi^E[1] \cup \Psi^E[2]$
$k = 1$
$\Psi^E[0] \cup \Psi^E[2]$
$k = 2$
$\Psi^E[0] \cup \Psi^E[1]$

(a) (b)

Figure 2-3: Example of the composite shape with $N_t = 2$. a) Shapes in effector frame. b) Composite shapes for $k = 0, 1, 2$.

constraint.

The SDF $\phi(\mathbf{p})$, defined with respect to a shape, is negative for points $\mathbf{p}$ inside the shape, positive for $\mathbf{p}$ outside, and zero for $\mathbf{p}$ on the boundary. $\phi(\mathbf{p})$ has magnitude equal to the shortest distance from $\mathbf{p}$ to the boundary of the shape. To enforce the condition for non-penetration, that *all* contact points be outside the object in the effector frame at *all* times, $\Psi^E[k], k = 0, ..., N_t$,

$$\phi[k](\mathbf{p}_i^E[k]) \geq 0, \forall i, k, \tag{2.2}$$

is imposed, where $\phi[k]$ is the SDFs of $\bigcup_{j \neq k} \Psi^E[j]$. $\Psi^E[k]$ is omitted from the non-penetration constraint at time k because contact point $\mathbf{p}_i^E[k]$ by definition lies on the boundary of $\Psi^E[k]$. Thus, if instead the SDF of $\bigcup_j \Psi^E[j]$ were used, the LHS of Eq. (2.2) would equal zero for all non-penetration-feasible solutions, resulting in a loss of gradient information. $\bigcup_{j \neq k} \Psi^E[j]$ is referred to as the *composite shape* at time k, and is visualized in Fig. 2-3.

There are several alternative options to this method. Non-penetration for each time could be left separate, with a separate constraint on the SDF of each $\Psi^E[k]$. Alternatively, the minimum over the SDFs of the individual $\Psi^E[k]$ could be used in place of the SDF of the composite shape. Both of these options, however, are more numerically challenging in a gradient-based framework. Eq. (2.2) consolidates the information from the separate $\Psi^E[k]$ and provides information about the depth of points within the composite shape rather than just the individual shapes, which yields fewer local minima. This idea is illustrated in Fig. 2-4.

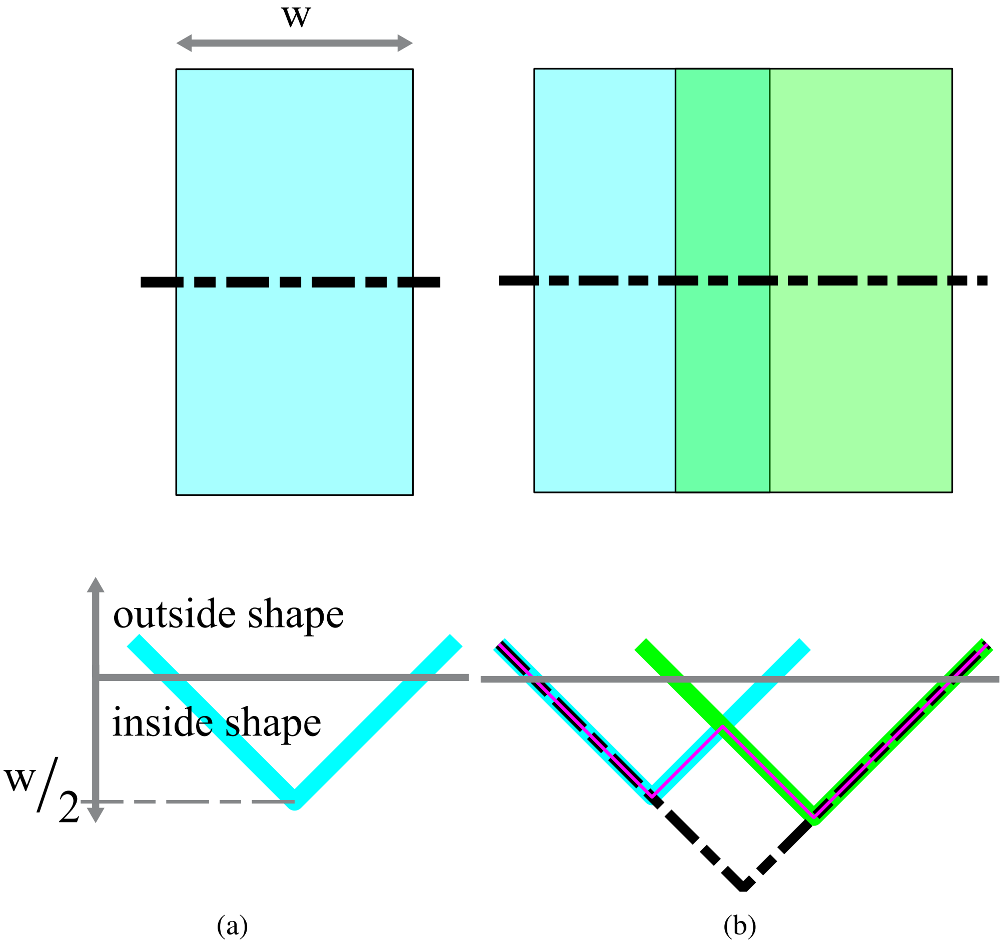


Figure 2-4: a) Top: rectangular shape, with black dashed line indicating where the SDF in the bottom figure is sampled. Bottom: rectangle SDF along slice from top figure. The SDF crosses 0 at the two rectangle boundary points on the sampling slice, is positive outside the shape, and is negative inside. The minimum SDF value, equal to half the rectangle width, is attained at the center of the rectangle. b) Top: two overlapping rectangles, with black dashed line indicating where the SDFs in the bottom figure are sampled. Bottom: Blue/green: individual SDFs of correspondingly colored shapes. Magenta: minimum over individual SDFs. Black dashed: SDF of union of rectangles. These three representations agree on which points are outside of both rectangles, but the SDF of the union has the simplest gradient information.

2.4 The optimization problem

As developed in Secs. 2.2 and 2.3, the constraints can be posed as functions of the effector motion alone. This makes it possible to avoid parameterizing the effector geometry and simply optimize over motion variables. The optimized effector motion, $\mathcal{P}^*, \Theta^*, \mathcal{S}^*$, is solved for, as defined below:

$$
\begin{aligned}
(\mathcal{P}^*, \Theta^*, \mathcal{S}^*) := \underset{\mathcal{P},\Theta,\mathcal{S}}{\arg\min} \quad & J(\mathcal{P}, \Theta, \mathcal{S}) && && \text{(2.3a)} \\
\text{s.t.} \quad & \mathbf{p}_i^E[k] - \mathbf{p}_i^E[k-1] = \frac{s_i[k]}{2}\left(\mathbf{d}_i^E[k-1] + \mathbf{d}_i^E[k]\right), && \forall i, k, k \neq 0 && \text{(2.3b)} \\
& \phi[k](\mathbf{p}_i^E[k]) \geq 0, && \forall i, k && \text{(2.3c)} \\
& \phi_\Omega(\mathbf{p}_i^E[k]) \geq 0, && \forall i, k && \text{(2.3d)} \\
& |\mathbf{p}_E^W[k] - \mathbf{p}_E^W[k-1]| \leq h[k]\dot{\mathbf{p}}_{max}, && \forall k, k \neq 0 && \text{(2.3e)} \\
& |\theta_E[k] - \theta_E[k-1]| \leq h[k]\dot{\theta}_{max}, && \forall k, k \neq 0 && \text{(2.3f)}
\end{aligned}
$$

Equality constraint (2.3b) is continuity, and inequality constraint (2.3c) is non-penetration. Inequality constraint (2.3d) is an optional addition for non-penetration of obstacles, where $\phi_\Omega(\mathbf{p})$ is the SDF of $\bigcup_j \Omega^E[j]$, the union of obstacles represented in the effector frame. Finally, because the intention is to approximate continuous motion, the effector position and orientation are constrained to not change between timesteps by more than some selected rates of change, $\dot{\mathbf{p}}_{max}$ and $\dot{\theta}_{max}$, multiplied by the timestep $h[k]$. Inequality constraints (2.3e) and (2.3f) impose this.

As an optional alternative to the formulation in Problem (2.3), the effector motion can be parameterized by $N_d - N_c$ optimization variables at each time, and the continuity constraint, Eq. (2.1), can be used to solve for the remaining effector state variables. This removes the continuity constraint from Problem (2.3) and reduces the number of optimization variables. This is useful when it is challenging to find a continuity-feasible solution. In particular, this is done for the problems with $N_d = 3$, $N_c = 2$ in Sec. 2.5.

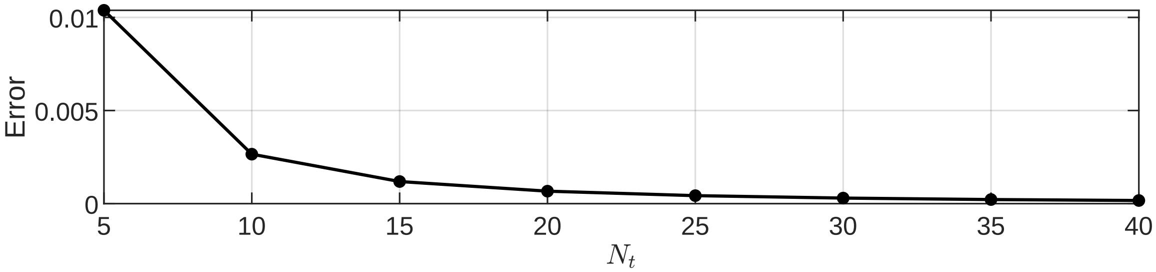


Figure 2-5: Error in solution found for the rock-climbing cam problem, plotted against number of timesteps. Error is defined as the average distance between discretized effector points in the solution found with the proposed framework and in the analytical solution, non-dimensionalized by the total horizontal travel distance of the wall (the same for all N_t).

2.5 Examples

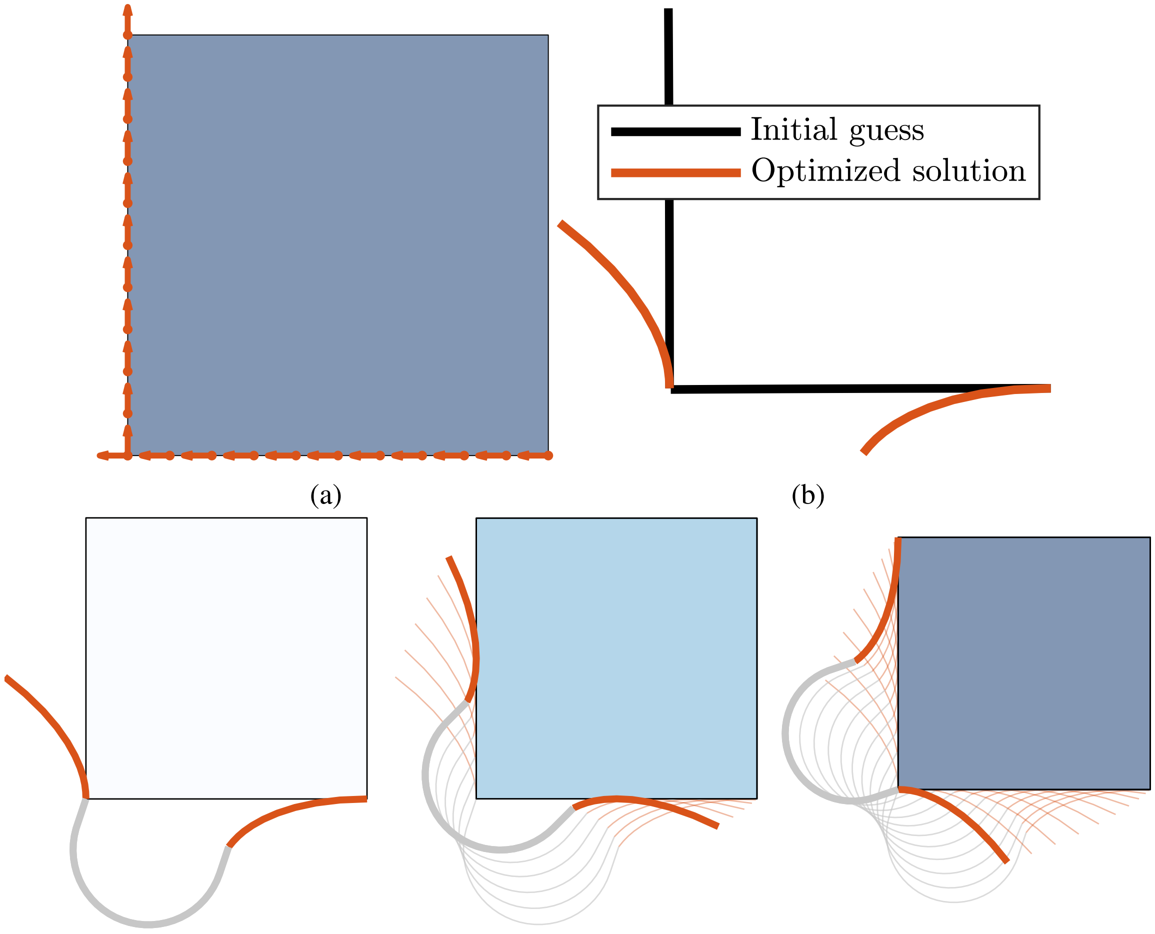
Here, results are presented for example problems. The problems already introduced continue to be discussed, and two new tasks are introduced: scooping a disk off a table, and opening jars with varying lid sizes. NLPs are solved using SNOPT [19,20] and MATLAB's `fmincon` [36].

2.5.1 Rock-climbing cam

The rock-climbing cam problem, introduced in Sec. 2.1.2, is fully constrained, with $N_d = N_c = 1$. The unique solution is a logarithmic spiral [43]. Fig. 2-5 compares solutions found with the proposed method to the analytical solution. Even with coarse discretizations, the solutions match closely, and, as expected, finer time discretization results in closer agreement.

2.5.2 Square slide

Fig. 2-2 shows a simple problem that illustrates the continuity constraint. Here, the properties of a similar problem, shown in Fig. 2-6a, are explored. Over the timespan of the problem, one contact point slides from the bottom-right corner of the square to the bottom-left corner, and a second contact point slides from the bottom-left corner to the top-left corner of the square. Specifying only the rotation angle of the effector at each time fully



(a)

(b)

(c)

Figure 2-6: a) Object and contact constraints for the square slide problem in world frame. The object is stationary, so we see only one object instance in this image. b) Initial guess and optimized solution using the cost function J_d. c) Usage of the optimized effector.

determines the effector state relative to that of the previous time, as $N_d - N_c = 1$. Only $\theta_E[1], ..., \theta_E[N_t]$ are optimized over. With three effector DOFs, the effector contact surface can arbitrarily be anywhere in the effector frame. Therefore $\mathbf{p}_E^W[0] = [0, 0]^T, \theta_E[0] = 0$ is set without loss of generality.

A stationary, L-shaped effector is actually a solution. If the effector undergoes no motion, the objects and constraints in the effector frame appear as in Fig. 2-6a, in which continuity and non-penetration are respected. However, instead of contacting just the contact point specified for each time, the resulting effector contacts every contact point at every time. Depending on the application, this may not be a desirable outcome. This inspires a

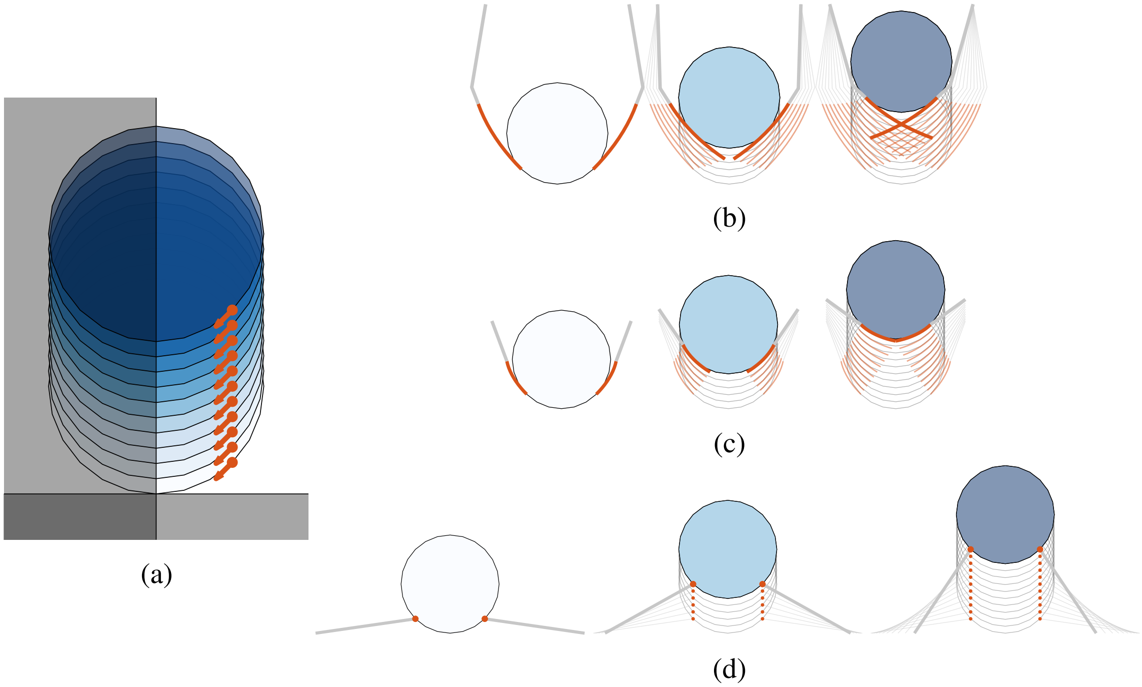


Figure 2-7: a) Object and contact constraints for the pickup problem in world frame. Gray regions represent obstacles (cropped). b-d) Solutions. Gray lines connect the effector surface to the center of rotation. b) and c) are solutions to the pickup problem with θ and vertical DOFs, minimizing J_y, without and with the vertical obstacle, respectively. d) is the solution to the pickup problem with θ and horizontal DOFs, minimizing J_s.

cost function that penalizes unintended proximity to the object:

$$J_d := -\sum_{k=0}^{N_t} \phi[k](\mathbf{p}_i^E[k]).$$

Minimizing this cost function maximizes the signed distances from contact points to the composite shapes. Figs. 2-6b and 2-6c show a resulting solution that achieves the goal of this user-specified cost function: The effector maintains greater distance from the object, except at the intended contact points, at all times.

2.5.3 Pickup

The goal of the pickup problem is to lift a disk from a surface, with two fingers contacting the disk symmetrically with contact-invariant geometry (the contact point remains the same relative to the object over time). The fingers may pivot, and translate vertically. A 24-sided polygon is used to approximate a disk.

Using symmetry, just one finger is designed and then mirrored for the full solution. Fig. 2-7a depicts the input for the problem, in which obstacles prevent the effector from penetrating the ground, and prevent the symmetric fingers from penetrating each other.

Limiting vertical motion is desirable in scenarios where space is limited. Rodriguez and Mason [44] solve a similar problem without the vertical-direction DOF, but produce a solution where the two fingers penetrate each other. A cost to minimize vertical movement is used,

$$J_y := \sum_{k=1}^{N_t} \left(\left(\mathbf{p}_E^W[k] - \mathbf{p}_E^W[k-1] \right) \cdot \hat{\jmath} \right)^2,$$

where $\hat{\jmath}$ is a world-frame vertical-direction unit vector. Indeed, removal of the vertical obstacle results in a solution without vertical-direction movement of the effector base, shown in Fig. 2-7b. Including the vertical obstacle results in the solution shown in Fig. 2-7c, where the fingers no longer penetrate each other, but a small amount of vertical translation occurs.

Instead of allowing vertical effector motion, by allowing horizontal motion and minimizing a measure of the length of the effector surface,

$$J_s := \sum_{i=0}^{N_c-1} \sum_{k=1}^{N_t} s_i[k]^2, \tag{2.4}$$

a simple solution is found: a single-point effector, shown in Fig. 2-7d.

2.5.4 Jar opener

The specification for the jar opener problem, introduced in [43], is shown in Fig. 2-8a. The goal is to create a versatile jar-opening tool that contacts jar lids all with the same contact geometry regardless of their radii. Like in Sec. 2.5.2, all DOFs are allowed and $\mathbf{p}_E^W[0] = [0, 0]^T, \theta_E[0] = 0$. With $N_d - N_c = 1$, the effector state is parameterized by $s_0[k]$, which is conducive to the objective functions used with this problem.

Suppose the goal is to use as little material as possible to generate this tool. In this case, we can minimize the total effector length using the cost function J_s defined in Eq. (2.4). With this cost function, a simple and intuitive solution is found, shown in Fig. 2-8b.

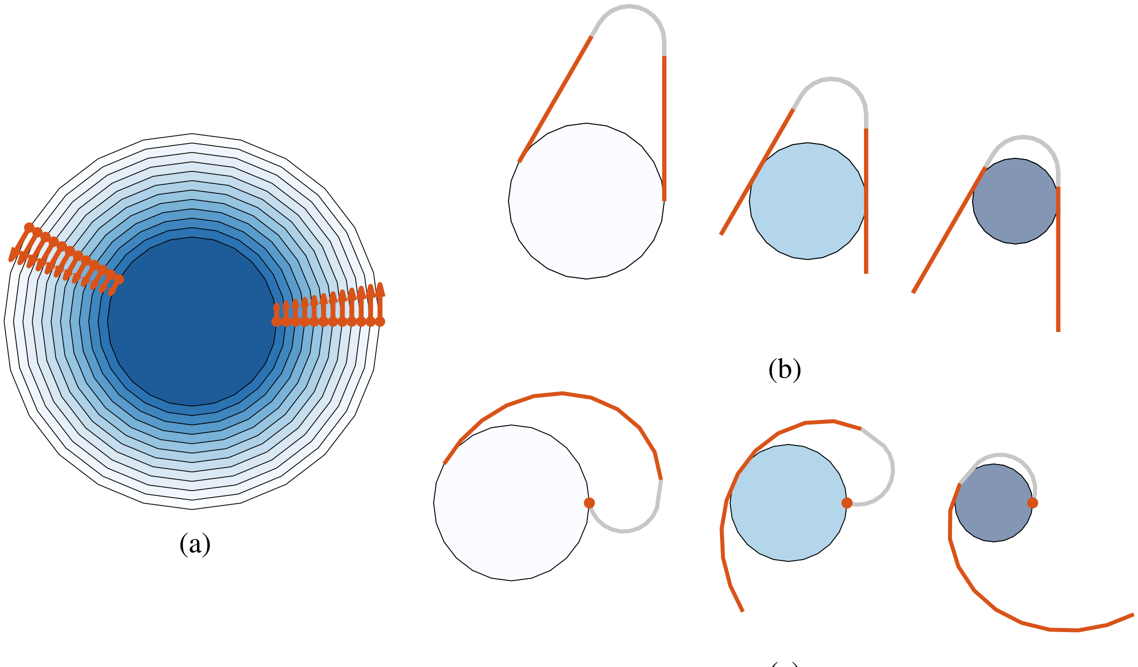


Figure 2-8: Color scheme is as defined in Sec. 1.4, except here instead of time, the object shape evolves (jar lid radius decreases) as k progresses. a) Object and contact constraints for the jar opener problem in world frame. b-c) Solutions using the cost functions J_s and J_{s_0} respectively.

Alternatively, suppose we use an expensive, high-quality material on one side of the tool for its frictional properties, so we wish to minimize the length of only that side. Using a similar cost function,

$$J_{s_0} := \sum_{k=1}^{N_t} s_0[k]^2,$$

the solution shown in Fig. 2-8c is found, where all contact points on one side converge to a single point on the effector.

These two solutions for the jar opener are qualitatively quite different, but both make sense for their respective cost functions. A range of solutions between the two extremes are feasible, but are not optimized for either of these costs.

2.5.5 Screwdriver

In reality, sometimes the only objective is to find a feasible effector. Arbitrary cost functions can harm the solution by pushing it toward the boundaries of the feasible set, which

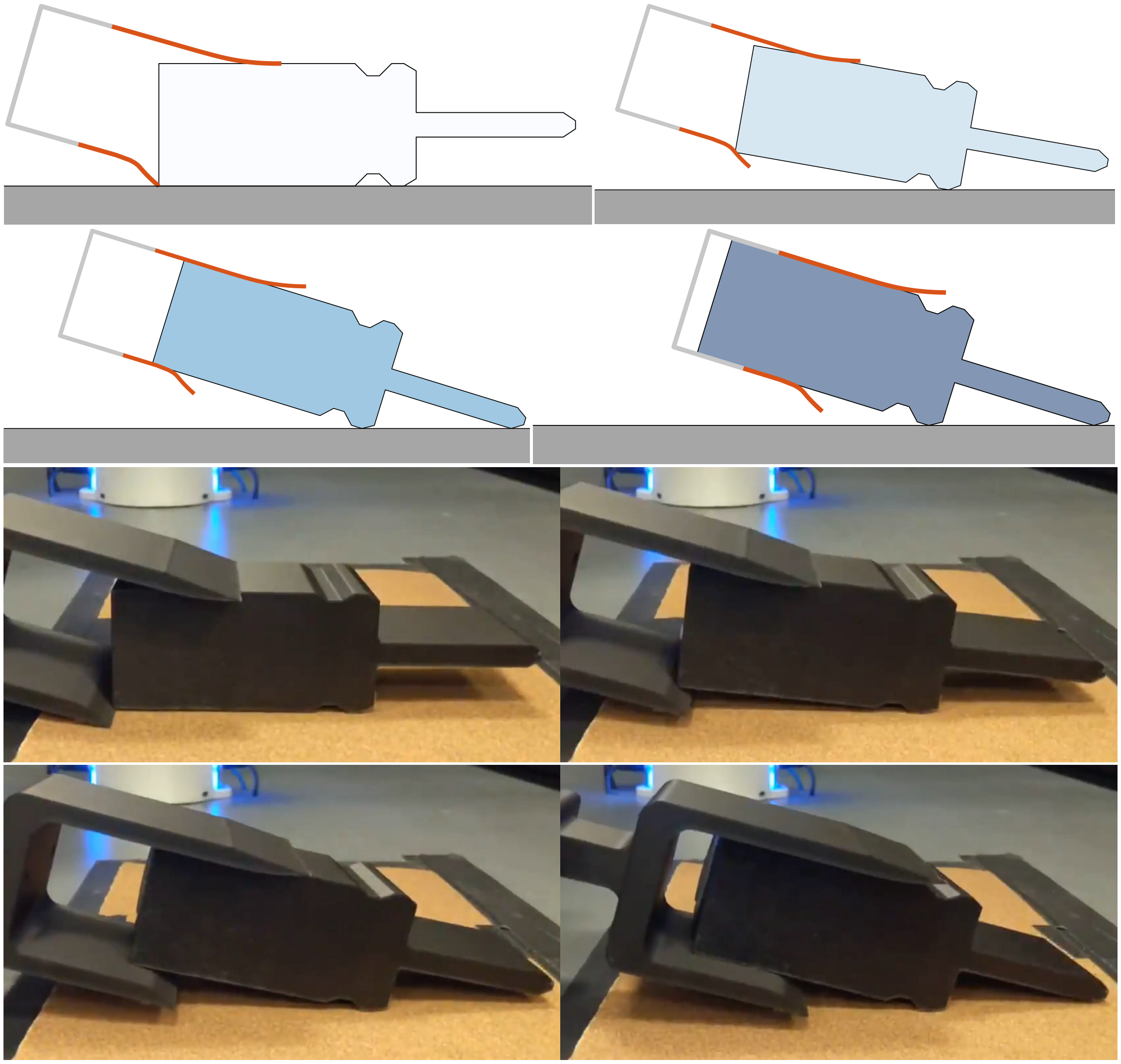


Figure 2-9: Top: solution to the screwdriver problem. Bottom: Robot demonstration of the solution to the screwdriver problem.

can cause edge cases that are not desired if this cost function was not selected deliberately. In the screwdriver problem, introduced in Sec. 1.2, a feasible solution to a complex problem is found, without focusing on optimality. The constant cost function

$$J_{const} := 1$$

is used in order to use the optimizer for constraint satisfaction. This problem requires two effector surfaces whose precise forms work in concert with one another in two phases of motion: scooping and sliding. The effector scoops a screwdriver off a table by the corner

Problem	N_t	N_d	N_c	N_p	$\dot{\mathbf{p}}'_{max}$	$\dot{\theta}'_{max}$	Solver	$T_s(s)$
square slide	10	3	2	1	0.5	30	`fmincon`	98.6
pickup 1	10	2	1	2	0.21	20	`fmincon`	13.3
pickup 2	10	2	1	2	0.21	20	`fmincon`	58.8
pickup 3	10	2	1	2	0.21	20	`fmincon`	4.8
jar opener 1	10	3	2	1	0.05	10	SNOPT	44.6
jar opener 2	10	3	2	1	0.05	10	SNOPT	22.2
screwdriver	20	3	2	1	0.06	5	`fmincon`	1.3

Table 2.1: Computational parameters for the example problems in Sec. 2.5. Here, N_p is the number of parameters used to represent motion at each time (see explanation of optional motion parameter reduction in Sec. 2.4). $\dot{\mathbf{p}}'_{max}$ is $\dot{\mathbf{p}}_{max}$ nondimensionalized by L/h, where L is the maximum object bounding box dimension. $\dot{\theta}'_{max}$ is $\dot{\theta}_{max}$ nondimensionalized by h. T_s is the time taken to solve the problem. "pickup 1" and "pickup 2" refer to the vertical-DOF pickup problems, without and with the vertical obstacle respectively. "pickup 3" refers to the horizontal-DOF pickup problem. "jar opener 1" and "jar opener 2" refer to the jar opener problem with cost J_s and J_{s_0} respectively.

of the handle, and then slides further onto the handle. The effector has full in-plane DOFs.

Like in Sec. 2.5.2, $\mathbf{p}_E^W[0] = [0, 0]^T, \theta_E[0] = 0$. For most of the trajectory, state is parameterized by $\theta_E[k]$ alone ($N_d - N_c = 1$). Toward the end, the contact tangents $\mathbf{d}_0^O[k]$ and $\mathbf{d}_1^O[k]$ are parallel, causing a degenerate case where $\theta_E[k]$ cannot be chosen freely but rather must be such that the object orientation remains constant in the effector frame. Code detects this condition and changes to parameterize by $s_0[k]$ for the remaining times.

The solution is shown in Fig. 2-9 (top). Intuitively, the bottom surface consists of a cupped feature for scooping, and then a flat surface for sliding in the later phase. The top surface consists of a curved feature that slides along the upper screwdriver surface, in harmony with the curvature of the lower surface, followed by a flat surface for the later phase. While the solution agrees with intuition, the precise shape of the effector could not have been designed with intuition alone.

2.6 Experiment

A demonstration was performed, shown in Fig. 2-9 (bottom), of the screwdriver pickup task with a robotic arm, on a high-friction surface. Models of the screwdriver and the effector geometry generated by the proposed framework were 3D printed and used for this experiment. Using compliance control to follow the trajectory generated by the proposed framework, the effector successfully lifts the screwdriver from the table.

2.7 Parameters and computation

The problem formulation scales well: The number of optimization variables is linear in N_t, and all presented problems are solved in under 100 seconds on a computer with Intel Core i7-10750H 2.60GHz CPU. Parameters used in the presented examples are shared in Table 2.1. The square slide problem solves slowly because the objective function involves evaluating SDFs. In contrast, the screwdriver problem is fast to solve because the reduced parameterization ensures that the continuity constraint is satisfied, and the cost is constant, meaning that the solver must only ensure that non-penetration is satisfied. Effector design is always performed offline, and not expected to be fast; therefore, computational efficiency has not been a focus, and the implementation could be further optimized for speed.

Several computational bottlenecks are of note. Firstly, unioning polygons is slow, and N_t polygons must be unioned $N_t + 1$ times per call of the constraint function, in order to find the composite shapes. Secondly, evaluating a signed distance is expensive, and must be performed $N_c \cdot (N_t + 1)$ times per call of the constraint function. These two issues may be jointly mitigated by pre-computing the SDF of each shape along a grid, and evaluating the SDF of the composite shape directly, as there exist algorithms for computing SDFs of unions of shapes [17].

Finally, SDFs are neither smooth nor convex, as discussed in Appendix A. SDFs can have sharp ridges along manifolds where points are equidistant from at least two closest boundary points. This can be challenging for NLP solvers and can lead to settling in local minima. This could be partially mitigated by smoothing the SDFs.

Chapter 3

Grasping

3.1 The grasp collection problem

While Chap. 2 discussed rigid effectors for contact trajectory satisfaction, this work has not yet been extended to effectors with internal DOFs, or problems where contact locations are free, and force balances must be considered. Both of these challenging extensions would increase the generality of the proposed framework. The present chapter introduces a preliminary extension to a grasping framework, in which the contact surfaces of parallel-jaw effectors are co-optimized with contact locations and effector configuration for grasping. The value of this work is primarily to demonstrate that contact location optimization, force balances, and internal effector DOFs can be incorporated into this NLP framework. The approach in this work optimizes over only contact location along a given polygonal object face, and leaves number of contacts, contact-face assignment, and contact-effector-jaw assignment optimization as a future extension, as discussed in Sec. 4.2.1.

The present grasping framework synthesizes effectors that can stably grasp every object in a set of objects input by the user. Designing a stable parallel-jaw grasp of a single object can often be done via intuition. However, designing an effector to stably grasp multiple objects of different geometries is more challenging. The framework from Chap. 2 provides a natural extension to consider multiple different objects; instead of indexing over time, the grasping problem indexes over objects. Constraints for non-penetration between object and effector are enforced for both effector jaws, for each object. This is done similarly to in Chap. 2, but instead of a single effector frame, there is an effector frame for each jaw. Unlike in Chap. 2, where contact locations are pre-specified, contact locations are optimized, and a grasp quality metric is considered to ensure the developed effectors lead to stable grasps. There is no notion of continuity, as there is no longer an approximation of continuous motion. Unlike in Chap. 2, without continuity, there is not a unique way to extract a global effector geometry from contact point locations; there remains freedom in how to do this. This work primarily focuses on ensuring that a feasible effector *can* be extracted, but briefly discusses potential algorithms for doing so, in order to reasonably visualize the effector results.

3.1.1 Problem formulation

The following items are required problem inputs:

- A set of polygonal objects, $\Psi^O[k]$, in the object reference frame (superscript O), at all object indices $k = 0, ..., N_t$.
- The orientation $\theta_O[k]$ and position $\mathbf{p}_O^W[k]$ of each object (subscript O) in the world frame (W).
- An object edge assignment, $v_i[k]$, for all contacts $i = 0, ..., N_c[k] - 1$, where $N_c[k]$ is the number of contacts on object k, and a set of contact indices $M_j[k] := \{i|$ contact i on object k belongs to jaw $j\}$ for each jaw, for all objects.

If a solution is found, the framework outputs:

- Effector configurations: $\mathbf{p}_E^W[k]$, $\theta_E[k]$, and $\gamma[k]$, the position, orientation, and jaw opening distance of the effector (E), for grasping each object. $\mathcal{P} := \{\mathbf{p}_E^W[0], ..., \mathbf{p}_E^W[N_t]\}$, $\Theta := \{\theta_E[0], ..., \theta_E[N_t]\}$, $\Gamma := \{\gamma[0], ..., \gamma[N_t]\}$.
- Contact positions along edges: $t_i[k]$, the ith contact's position distance from an edge vertex, normalized by the edge length, for all contacts on all objects. $\mathbf{t}[k] := [t_0[k], ..., t_{N_c[k]-1}[k]]^T$, and $\mathcal{T} := \{\mathbf{t}[0], ..., \mathbf{t}[N_t]\}$.
- Two polygons, each representing the shape of an effector jaw.

As in Chap. 2, the approach relies on expressing variables in the effector frames. As the parallel-jaw gripper has an internal DOF, let E_0 and E_1 denote the two effector frames, and let $\Psi^{E_j}[k]$ and $\mathbf{p}_i^{E_j}[k]$ denote object k and position of the ith contact point on object k, represented in E_j, $j \in \{0, 1\}$.

3.1.2 Approach

At each iteration while solving the NLP, a candidate solution, $\{\mathcal{P}, \Theta, \Gamma, \mathcal{T}\}$, is evaluated for effector-object penetration, and for grasp stability. The non-penetration constraint is assessed similarly to in Sec. 2.3, extended in Sec. 3.3 via the notion that the effector jaws approach each object on a linear trajectory (the motion of the jaw closure).

Grasp stability is assessed via a quadratic program (QP), described in Problem (3.1). A grasp quality metric is defined, based on the optimal solution to this QP, and enters the NLP objective function. As the QP constraint matrices are functions of the NLP variables, the QP is re-solved at each NLP iterate. An alternative option to explicitly solving the QP is to merge the QP variables and constraints into the NLP variables and constraints. The advantage of explicitly solving the QP is that the global optimum is guaranteed. While it would be possible to include the Karush–Kuhn–Tucker (KKT) conditions in the NLP constraints to ensure that any feasible NLP solution attains the QP global optimum, there would be no guarantee of finding a feasible solution to this program, even if one exists. Furthermore, the complementary slackness KKT condition is not smooth, which could lead to numerical issues in solving the NLP.

The NLP is nonconvex and the geometry of the feasible space makes the program difficult to solve. In particular, as discussed in Sec. 3.3, the non-penetration constraint makes it difficult for solvers to globally traverse the feasible space. This can be mitigated by running the NLP solver multiple times with randomized initial guesses. In particular, due to the nature of the nonconvexity discussed in Sec. 3.3, the initial guesses for the vertical coordinates of the effector positions are randomized. While repeated initial guess randomization has been chosen for simplicity, other methods such as simulated annealing [32] or particle swarm optimization [30] could yield improvements.

The NLP constraints and objective encourage the conditions on the contact points and effector configurations that *allow* a feasible effector with good grasps of the input set of objects. However, after the NLP solver terminates there remains freedom in extracting the effector geometry based on the NLP solution. Selecting the effector shape based on the NLP solution is not the focus of this work, but the following is one simple method that is used for visualizing results in Sec. 3.5. For each jaw j:

1. Sort $\mathbf{p}_i^{E_j}[k]$ for all k and $i \in M_j[k]$ by vertical coordinate.
2. Add to this set any other points desired on the effector, e.g., horizontally offset points to give width to the effector jaw.
3. Form a polygon from these points by connecting consecutive points with line segments.
4. Subtract the composite shape, $\bigcup_k \tilde{\Psi}^{E_j}[k]$, explained in Sec. 3.3, from this polygon.

3.2 Grasp stability

3.2.1 Mathematical program formulation

In order to optimize effectors for grasping objects, it is necessary to be able to check whether a candidate grasp is stable. In particular, it is necessary to check whether the grasping configuration and contact placement allow the object to be in static equilibrium. Traditionally, this is approached by searching for contact forces that satisfy the equilibrium equations. However, as grasps are generally statically indeterminate, many sets of contact forces satisfy equilibrium equations, but would not arise passively. For example, consider grasping a square between two flat parallel-gripper jaws, with an external force pulling the square perpendicular to the jaw axes. There exist sets of contact forces that result in static equilibrium. For example, a solution where the jaws "squeeze" the square, creating normal forces large enough to support friction forces that resist the external force. However, it is clear that this set of forces does not arise unless the grasp is preloaded, i.e., the internal forces that "squeeze" the square must be applied by the effector. To resolve this issue, the proposed framework borrows two concepts from Haas-Hegar [25]: specifying preload in the grasping model, and a compliance model for resolving static indeterminacy. The analysis in the present section is for an individual object; grasp stability is considered independently for each k.

To specify preload in the grasp, the horizontal (jaw axis direction) force on the left jaw is constrained to equal some selected value F, via the following relation:

$$[1,0](\mathcal{J}[k](\theta_E,\mathbf{t}))^T\mathbf{c}-F=0,$$

where $\mathcal{J}[k](\theta_E,\mathbf{t})$ denotes the *hand matrix* [48] for object k, which is a function θ_E and $\mathbf{t}$ for a given object. The left jaw has been chosen arbitrarily to set the preload, as constraining the preload at both jaws would overconstrain the system.

The compliance model from Haas-Heger et al. [25] is used, in which contact normal forces are governed by a linear-spring compliance model, while tangent forces are not affected by the virtual springs, but are constrained to lie within their friction cones. Denote the normal and tangent forces at the ith contact $c_{i,n}, c_{i,t}$ respectively, and concatenate

$$\mathbf{c}:=[c_{0,n},c_{0,t},...,c_{N_c-1,n},c_{N_c-1,t}]^T.$$

The compliance model requires the introduction of displacements, $d_{i,n}$ and $d_{i,t}$, which indicate how far the virtual spring at the ith contact deforms in the contact normal and tangent directions respectively. These displacements must be made consistent with a rigid-body displacement, $\mathbf{r}$, of the object and jaw motions, $\mathbf{q}$, via the following relation [24]:

$$\mathbf{d}:=[d_{0,n},d_{0,t},...,d_{N_c-1,n},d_{N_c-1,t}]^T=(\mathcal{G}[k](\theta_E,\mathbf{t}))^T\mathbf{r}-\mathcal{J}[k](\theta_E,\mathbf{t})\mathbf{q},$$

where $\mathcal{G}[k](\theta_E,\mathbf{t})$ is the *grasp matrix* [48] for object k. The compliance model relation is simply [24]

$$c_{i,n}=-d_{i,n},\quad\forall i,$$

where a spring constant of unity has been used. This seemingly arbitrary selection serves only to scale the displacements, whose units are arbitrary.

Virtual spring normal displacements $d_{i,n}$ are constrained to be non-positive, signifying that the virtual springs can only compress (contact normal forces must be non-negative). For a given external wrench $\mathbf{w}$, and object index k, the grasp stability program, Problem (3.1), can be considered.

$$
\begin{aligned}
J_G^*[k](\theta_E, \mathbf{t}) := \min_{\mathbf{r},\mathbf{q},\mathbf{c}} \quad & J_G(\mathbf{r}, \mathbf{q}) && && \text{(3.1a)}\\
\text{s.t.} \quad & \mathcal{G}[k](\theta_E, \mathbf{t})\mathbf{c} + \mathbf{w} = \mathbf{0} && && \text{(3.1b)}\\
& c_{i,n} = -d_{i,n} && \forall i && \text{(3.1c)}\\
& d_{i,n} \leq 0 && \forall i && \text{(3.1d)}\\
& [1, 0](\mathcal{J}[k](\theta_E, \mathbf{t}))^T \mathbf{c} - F = 0 && && \text{(3.1e)}\\
& -\mu c_{i,n} \leq c_{i,t} \leq \mu c_{i,n} && \forall i. && \text{(3.1f)}
\end{aligned}
$$

where the objective function $J_G(\mathbf{r}, \mathbf{q})$ is developed in Sec. 3.2.2, and μ is the coefficient of friction. Inequality constraint (3.1b) gives the static equilibrium balance of the object, and inequality constraints (3.1f) enforce that the contact forces lie within their friction cones.

3.2.2 A grasp quality metric

This section motivates and defines a grasp quality metric, $J_G(\mathbf{r}, \mathbf{q})$, tailored for parallel-jaw grasps. The constraints in Problem (3.1) sufficiently prevent infeasible grasps. However, they allow grasps that are intuitively poor.

Fig. 3-1 shows two candidate grasps for a square object. Throughout this work, the axis of the jaws is assumed to be horizontal. In Fig. 3-1, the configuration on the left is an intuitively good way to grasp the square object. In particular, the axis of jaw action is aligned with contact normal directions, and contacts on either side of the square are vertically aligned. Thus, a firm grasp can be formed by tightening the jaws, without relying on friction forces in order to achieve a preloaded equilibrium in the absence of external wrenches. In contrast, in the right configuration in Fig. 3-1, the jaw axis has small components in the contact normal directions. Additionally, the large vertical distance between

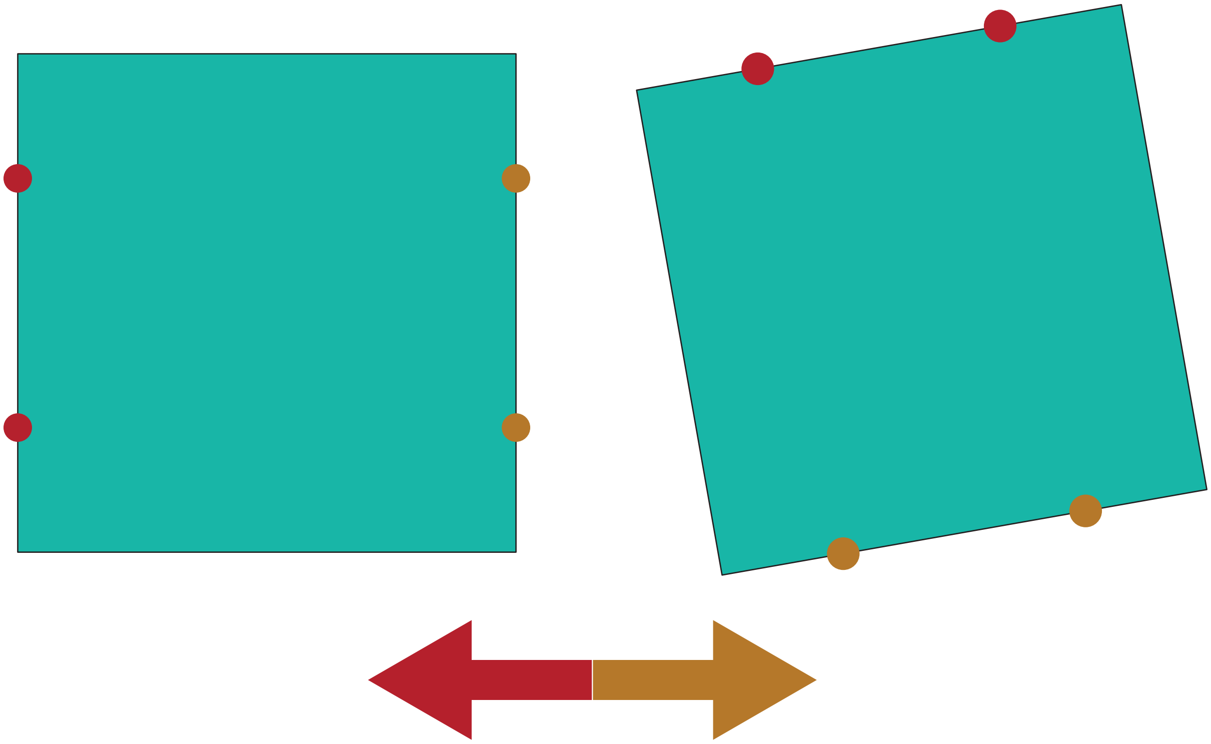


Figure 3-1: Left: An intuitively good grasp. Right: an intuitively bad grasp. The contact points are the same relative to the object, but the whole configuration is rotated by 80^o relative to the gripper, compared to the grasp on the left. The red/orange arrow reminds the reader of the horizontal axis of action of the parallel-jaw gripper.

the contacts on the left and right sides of the square risks a torque resulting from forces in the jaw axis direction at these points. However, this grasp is feasible under Problem (3.1), corresponding to large jaw displacements $\mathbf{q}$ that achieve compression of the virtual springs and thus normal contact forces. Generally, grasps that are worse-suited for achieving contact-normal preload correspond to larger displacements $\mathbf{r}$ and $\mathbf{q}$.

This observation motivates the grasp quality metric,

$$J_G(\mathbf{r}, \mathbf{q}) = \frac{1}{L^2}(\mathbf{r}^T\mathbf{r} + \mathbf{q}^T\mathbf{q}), \tag{3.2}$$

where $L[k]$ is a characteristic length, to nondimensionalize. In numerical experiments,

$$L[k] := \frac{1}{N_c[k]} \sum_{i=0}^{N_c[k]} ||\mathbf{p}_i^O[k]||,$$

the average contact point distance from the reference point of the object, is used.

Eq. (3.2) uses a 2-norm, as opposed to a 1-norm. While only the virtual spring compressions $\mathbf{d}$ relate the displacements $\mathbf{r}$ and $\mathbf{q}$ to the contact forces and therefore equilibrium, there are multiple values of $\mathbf{r}$ and $\mathbf{q}$ that lead to the same spring compressions. The spring compressions are equal to $B[\mathbf{r}^T, \mathbf{q}^T]^T$, where B is a matrix containing every other row of the matrix $[\mathcal{G}[k](\theta_E, \mathbf{t})^T, -\mathcal{J}[k](\theta_E, \mathbf{t})]$, mapping from displacements to spring compressions. Minimizing the 2-norm of $[\mathbf{r}^T, \mathbf{q}^T]^T$ ensures that there is no component in the nullspace of B, removing dependency on arbitrary differences of expression of displacements that are effectively the same. This 2-norm makes Problem (3.1) a quadratic program.

Solving Problem (3.1) at a particular θ_E, for a particular object index k, with contact locations $\mathbf{t}$ yields a J_G^* value. In Fig. 3-2, these values are plotted for four polygons with some selected contact placements. These polygons are perturbed versions of regular polygons, in order to prevent result dependency on precise features such as parallel edges or equal contact point distances. The shown slices of this metric display several desirable qualities in these plots: J_G^* increases rapidly when effector orientation θ_E approaches infeasible regions, J_G^* appears smooth and convex, and J_G^* is minimized at configurations that are intuitively good grasps. In experiments, $F = 1$ is used in equality constraint (3.1e), i.e., the force exerted in the jaw axis direction by the left jaw on the object is unity. As there are no specified units, this seemingly arbitrary selection dictates only that the preload is positive; for some scalar $\alpha \geq 0$, scaling $F \to \alpha F$ results in $J_G^*[k](\theta_E, \mathbf{t}) \to \alpha^2 J_G^*[k](\theta_E, \mathbf{t})$ without affecting feasibility.

3.3 Non-penetration

As in Sec. 2.4, non-penetration between effector and object is imposed via object SDFs. As the parallel-jaw effector has an internal DOF, there is no longer a single effector frame E that can remain fixed to every part of the effector. Instead, there are two effector frames, E_j, where $j \in \{0, 1\}$ indexes over the two jaws. Non-penetration is enforced in both effector frames.

Grasps are assumed to be achieved via pure translation of both jaws along the jaw axis, and results must allow effectors that are free of collisions with the objects along this path.

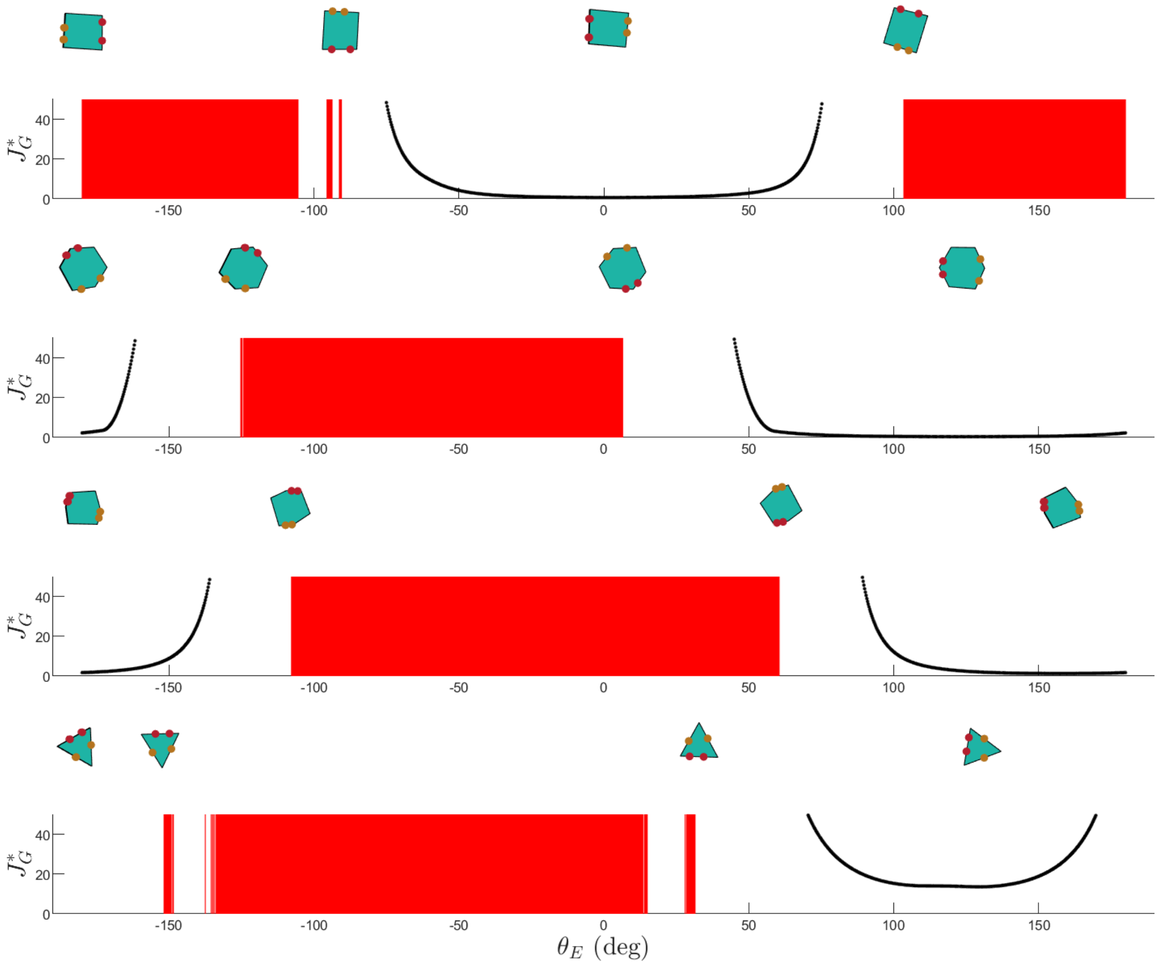


Figure 3-2: Optimized grasp stability metric J_G^* plotted as a function of θ_E for constant contact positions relative to object, for four perturbed polygons. Red regions indicate that the grasp is infeasible. The objects' orientations in the E-frame (no distinction between E_0 and E_1 because only rotation matters) are shown above significant transitions in J_G^* behavior over θ_E. The objects rotate clockwise relative in the E-frame as θ_E increases.

Equivalently, the objects must be able to be swept along this axis in the effector frames without being penetrated by any contact points. For example, in Figs. 3-3c and 3-3d, a contact point lies within this sweep, making the grasp infeasible. In contrast, Figs. 3-3a, 3-3b, 3-3e and 3-3f show the same object swept in the effector frames for two other θ_E values, which do not result in penetration. To represent this, each object in each effector frame $\Psi^{E_j}[k]$ is swept to form $\tilde{\Psi}^{E_j}[k]$, before forming the composite shape in E_j, $\bigcup_k \tilde{\Psi}^{E_j}[k]$, and enforcing

$$\tilde{\phi}_j(\mathbf{p}_i^{E_j}[k]) \geq 0, \quad \forall j, k, i \in M_j[k] \tag{3.3}$$

where $\tilde{\phi}_j$ is the SDF of $\bigcup_k \tilde{\Psi}^{E_j}[k]$. In non-penetration constraint (2.3c), $\phi[k]$, the SDF of $\bigcup_{j \neq k} \Psi^E[j]$, was used, as removing $\Psi^E[k]$ from the non-penetration constraint for $\mathbf{p}_i^E[k]$ helps improve constraint gradient information. In the present setting, however, as $\Psi^{E_j}[k]$ do not approximate a continuum over k, $\bigcup_{j \neq k} \tilde{\Psi}^{E_j}[j]$ may be disjoint, which can cause complications.

In the effector frames, these horizontally swept objects are wide obstacles for the contact points to avoid. Over the NLP solver iterates, in order for the vertical ordering of objects to change, Eq. (3.3) must become violated as the objects pass through each other vertically, before it can again be resolved. Thus, vertical reordering of the objects relative to the ordering in the initial guess is not expected, motivating the initial guess randomization over vertical coordinates explained in Sec. 3.1.2.

3.4 The optimization problem

The constraints and objective can be posed as functions of the effector configurations for all k ($\mathcal{P}$, Θ, and Γ), and the contact coordinates $\mathcal{T}$. The full NLP is formalized as Problem (3.4).

$$J^* := \min_{\mathcal{P},\Theta,\Gamma,\mathcal{T}} \quad wJ_P(\mathcal{P}, \Theta, \Gamma, \mathcal{T}) + \sum_{k=0}^{N_t} J_G^*[k](\theta_E[k], \mathbf{t}[k]) \tag{3.4a}$$

$$\text{s.t.} \quad \tilde{\phi}_j(\mathbf{p}_i^{E_j}[k]) \geq 0, \quad \forall j, k, i \in M_j[k] \tag{3.4b}$$

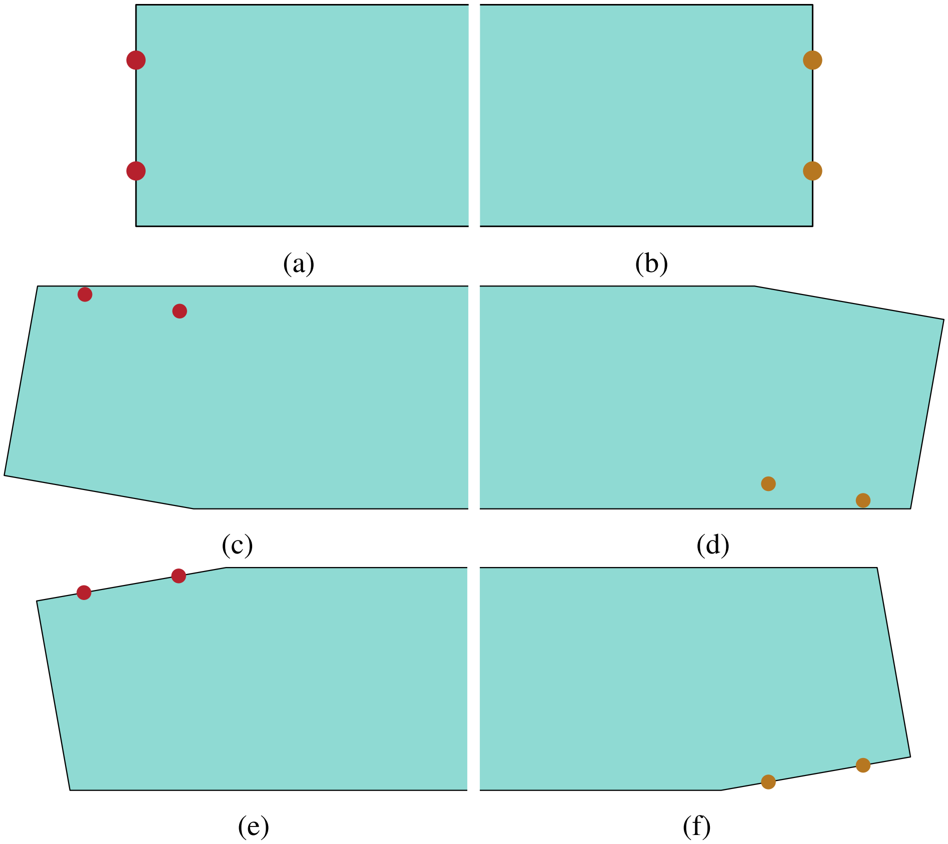


Figure 3-3: Effector-frame sweeps, $\tilde{\Psi}^{E_0}$ ((a), (c), (e)) and $\tilde{\Psi}^{E_1}$ ((b), (d), (f)), for a square object, with three different effector angles, $\theta_E = 0$ deg ((a), (b)), $\theta_E = 100$ deg ((c), (d)), and $\theta_E = 80$ deg ((e), (f)). The grasping configuration in (c) and (d) has penetration and is thus infeasible, as the contact points are within the object sweep in the effector frames.

where J_P is a user-specified cost function, and w is a scalar for tuning the relative weighting of the cost terms. Here, the grasp quality metric from Sec. 3.2.2 is incorporated in the NLP cost linearly, summing the values over all k. Note that for the set of Θ that define an infeasible grasp for at least one k, the second term in the cost (3.4a) is undefined. However, as noted in Sec. 3.2.2, $J_G^*[k](\theta_E, \mathbf{t}[k])$ increases rapidly as its arguments approach infeasible values for object index k. Therefore, with Problem (3.4) being solved in a gradient-based scheme with a reasonable step size, if the solver is initialized with an assignment that leads to feasible grasps, in practice it remains feasible over the iterates. The initial guess need not be non-penetration-feasible. Finding an initial guess that leads to feasible grasps is typically easy to do via intuition.

3.5 Examples

This section discusses results obtained by the presented framework for several different example problems and cost functions, and slight variations to the framework. As the NLP solver is repeatedly run with randomized initial guesses, multiple results are obtained. The five best-cost solutions are presented for each example problem.

3.5.1 Compact grippers

A potential reason to design a single parallel-jaw gripper that can grasp multiple objects is that it is desirable to multipurpose features of the effector. This goal may be motivated by conserving effector material or space limitations. Results in this section optimize a simple cost function that encourages the contact points to be near each other in the effector frames, effectively encouraging the necessary effector contact surface to be compact:

$$J_{PC} := \sum_{j=0}^{1} \sum_{k=0}^{N_t} \sum_{i \in M_j[k]} ||\mathbf{p}_i^{E_j}[k] - \bar{p}_j||^2,$$

where $\bar{p}_l$ is the centroid of points in E_j,

$$\bar{p}_j := \frac{1}{N_t + 1} \sum_{k=0}^{N_t} \sum_{i \in M_j[k]} \mathbf{p}_i^{E_j}[k].$$

The external wrench is set to $\mathbf{w} = [0, 0, 0]^T$ in these experiments. Figs. 3-4 to 3-8 show results for this example problem with four slightly perturbed regular polygon objects.

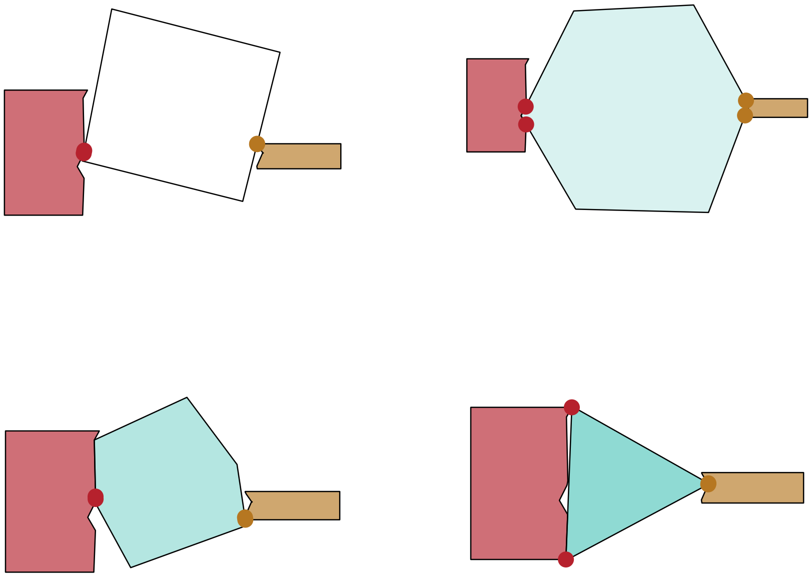


Figure 3-4: Best-cost compact gripper solution for convex shapes.

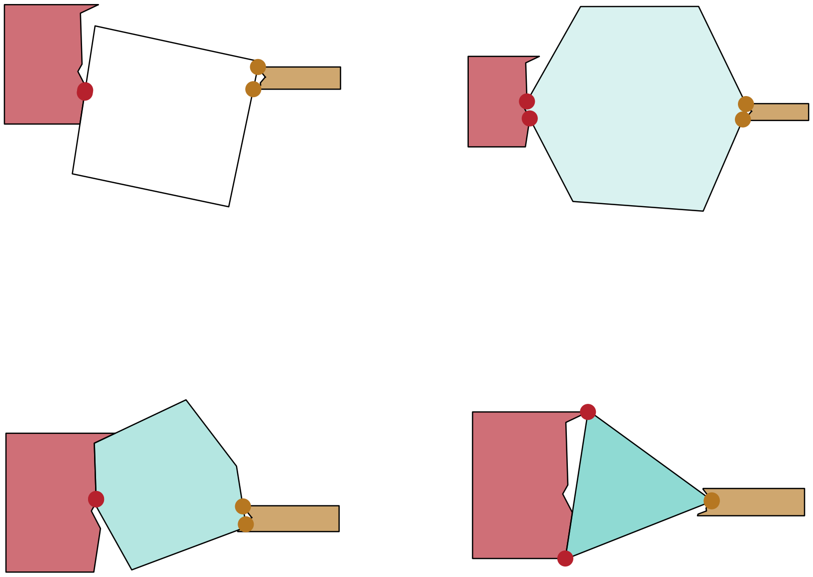


Figure 3-5: Second-best-cost compact gripper solution for convex shapes.

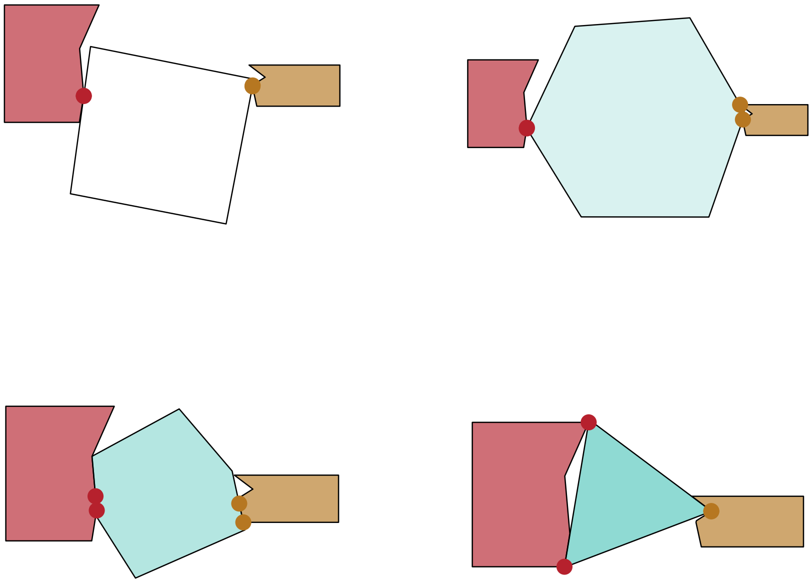


Figure 3-6: Third-best-cost compact gripper solution for convex shapes.

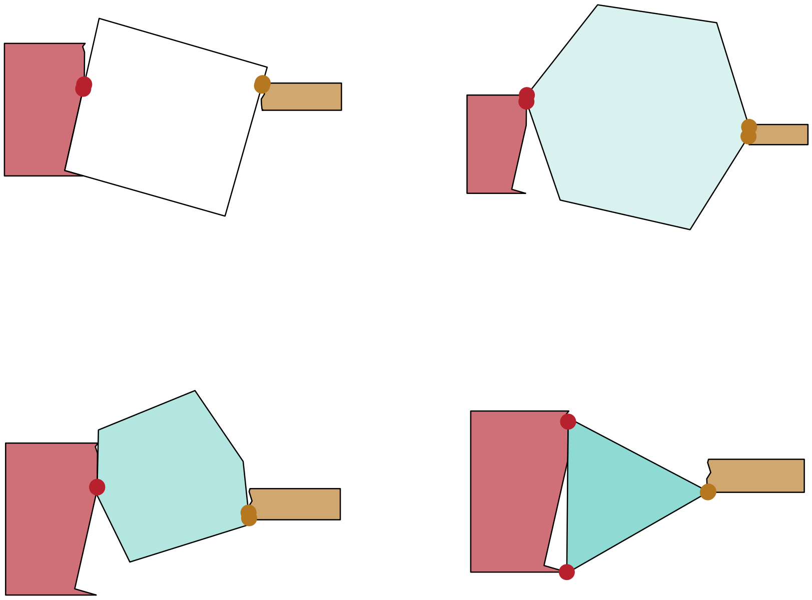


Figure 3-7: Fourth-best-cost compact gripper solution for convex shapes.

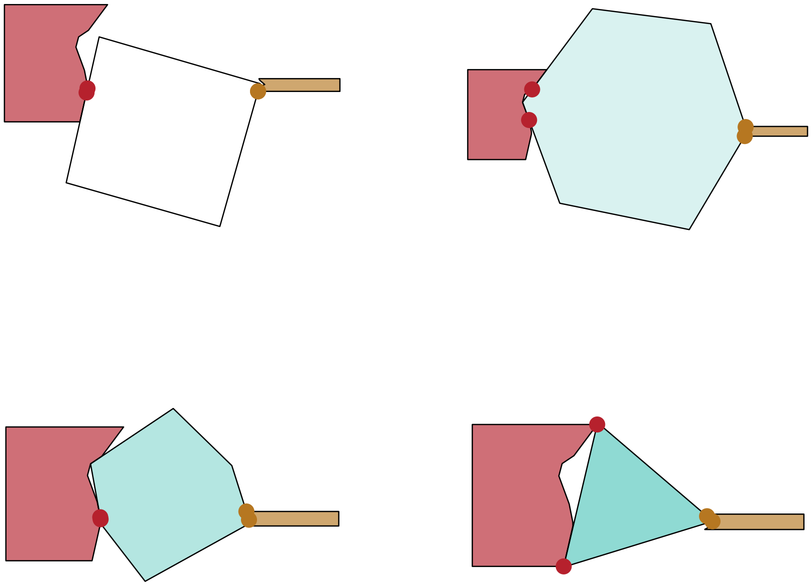


Figure 3-8: Fifth-best-cost compact gripper solution for convex shapes.

In these results, as grasp quality has been assessed with respect to zero external wrench, some results are clearly not tolerant to exernal torques. Note how contact points tend to bunch up on the square and smaller hexagons, resulting in low ability to react torques. This shortcoming motivates a modification to the formulation, discussed in Sec. 3.5.2.

3.5.2 Modifications for robustness to external torques

While $\mathbf{w} = [0, 0, 0]^T$ has been used until now, assigning $\mathbf{w}$ in Problem (3.1) is a viable way to evaluate grasps with respect to an external wrench. However, the correct choice of $\mathbf{w}$ is not obvious. This section explains why, in fact, evaluating grasps with respect to $\mathbf{w} = [0, 0, 1]^T$ and $\mathbf{w} = [0, 0, -1]^T$ is a reasonable choice.

As the contacts are assumed to be frictional, and the grasp is preloaded, the grasps generated are automatically tolerant to external forces. Therefore, in lieu of particular knowledge about actual intended loading, it is reasonable to select $\mathbf{w}$ to evaluate grasps with respect to only torques. Consider Problem (3.1) and suppose now that F is free. Then, for some scalar $\alpha \geq 0$, scaling $\mathbf{w} \to \alpha\mathbf{w}$ results in $J_G^*[k](\theta_E, \mathbf{t}) \to \alpha^2 J_G^*[k](\theta_E, \mathbf{t})$ without affecting feasibility. Therefore, this modified quadratic program, Problem (3.5), with $\mathbf{w} = [0, 0, 1]^T$ or $\mathbf{w} = [0, 0, -1]^T$, can be used to evaluate a grasp with respect to any positive or negative torque respectively.

$$
\begin{aligned}
J_G^*[k](\theta_E, \mathbf{t}, \mathbf{w}) := \min_{\mathbf{r},\mathbf{q},\mathbf{c},F} \quad & J_G(\mathbf{r}, \mathbf{q}) && && \text{(3.5a)} \\
\text{s.t.} \quad & (\mathcal{G}[k](\theta_E, \mathbf{t})\mathbf{c} + \mathbf{w} = \mathbf{0} && && \text{(3.5b)} \\
& c_{i,n} = -d_{i,n} && \forall i && \text{(3.5c)} \\
& d_{i,n} \leq 0 && \forall i && \text{(3.5d)} \\
& [1, 0](\mathcal{J}[k](\theta_E, \mathbf{t}))^T \mathbf{c} - F = 0 && && \text{(3.5e)} \\
& -\mu c_{i,n} \leq c_{i,t} \leq \mu c_{i,n} && \forall i. && \text{(3.5f)}
\end{aligned}
$$

The resulting objective function to replace cost (3.4a) is

$$
J = wJ_P(\mathcal{P}, \Theta, \Gamma, \mathcal{T}) + \sum_{k=0}^{N_t} \left(J_G^*[k](\theta_E[k], \mathbf{t}[k], [0, 0, 1]^T) + J_G^*[k](\theta_E[k], \mathbf{t}[k], [0, 0, -1]^T) \right).
$$

Figs. 3-9 to 3-13 show results for this modified formulation with the four slightly perturbed regular polygon objects, again using the cost function J_{PC} from Sec. 3.5.1.

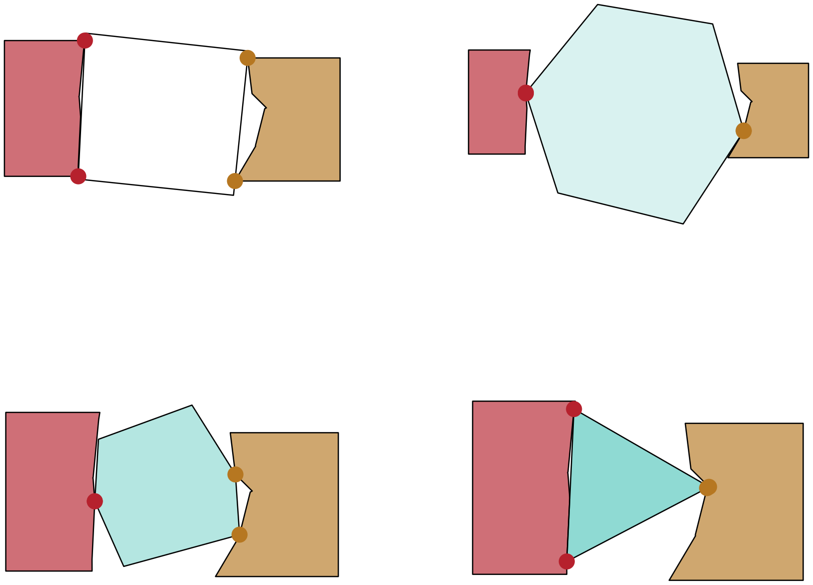


Figure 3-9: Best-cost compact, torque-robust gripper solution for convex shapes.

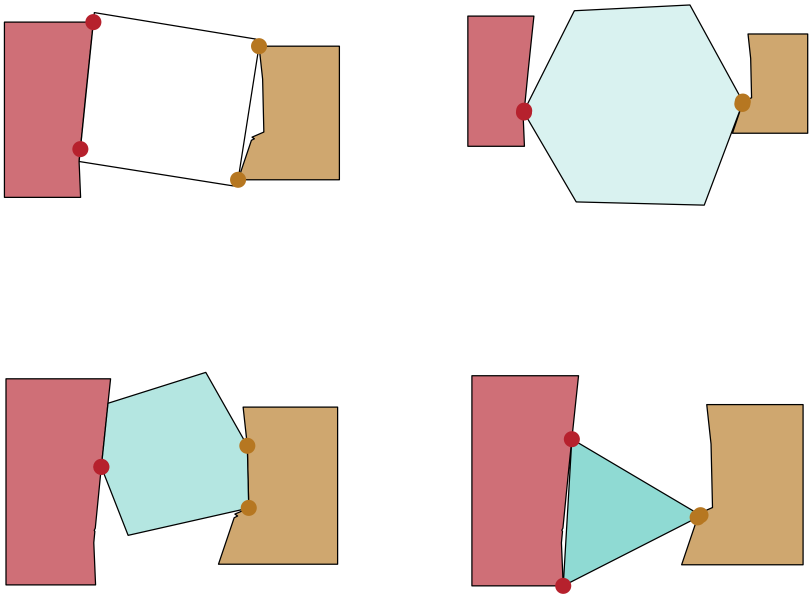


Figure 3-10: Second-best-cost compact, torque-robust gripper solution for convex shapes.

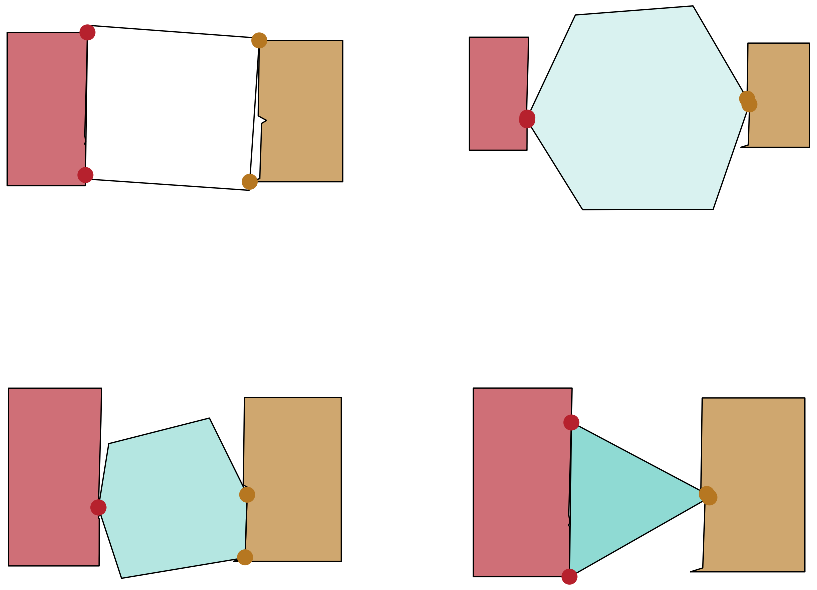


Figure 3-11: Third-best-cost compact, torque-robust gripper solution for convex shapes.

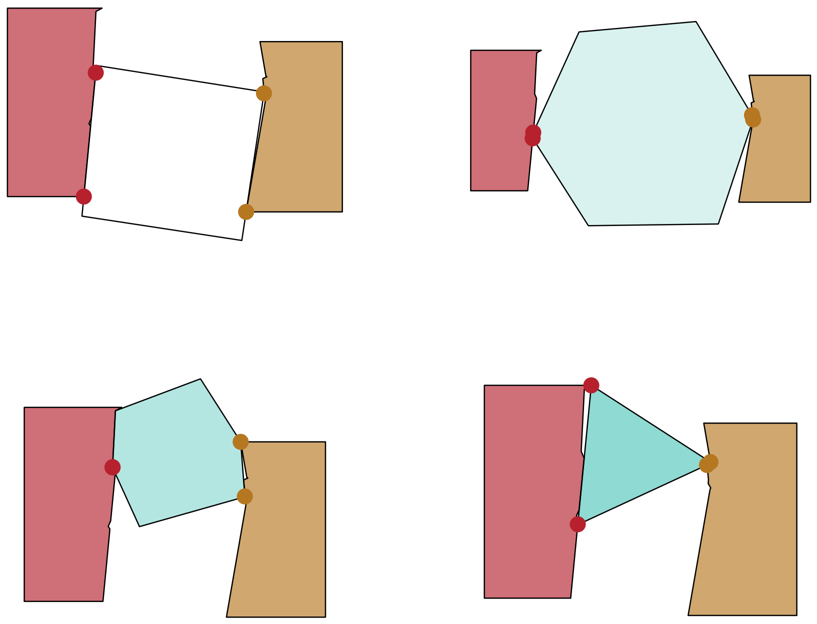


Figure 3-12: Fourth-best-cost compact, torque-robust gripper solution for convex shapes.

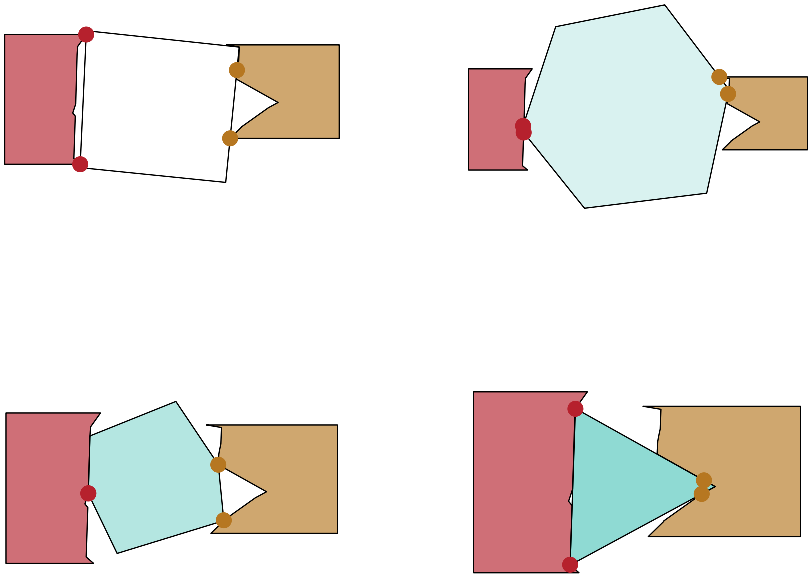


Figure 3-13: Fifth-best-cost compact, torque-robust gripper solution for convex shapes.

Indeed, the contact points on the square and small hexagon move away from each other, providing more torque resilience in the resulting grasps. While the contacts on the large hexagon remain near each other, the contact normal directions are such that the contacts near each other are not redundant, and the grasp is able to react torques.

With this successful result, let us now study a more complex set of objects using the same problem formulation and same cost function. Figs. 3-14 to 3-18 show results for a set of five objects with significant nonconvexities.

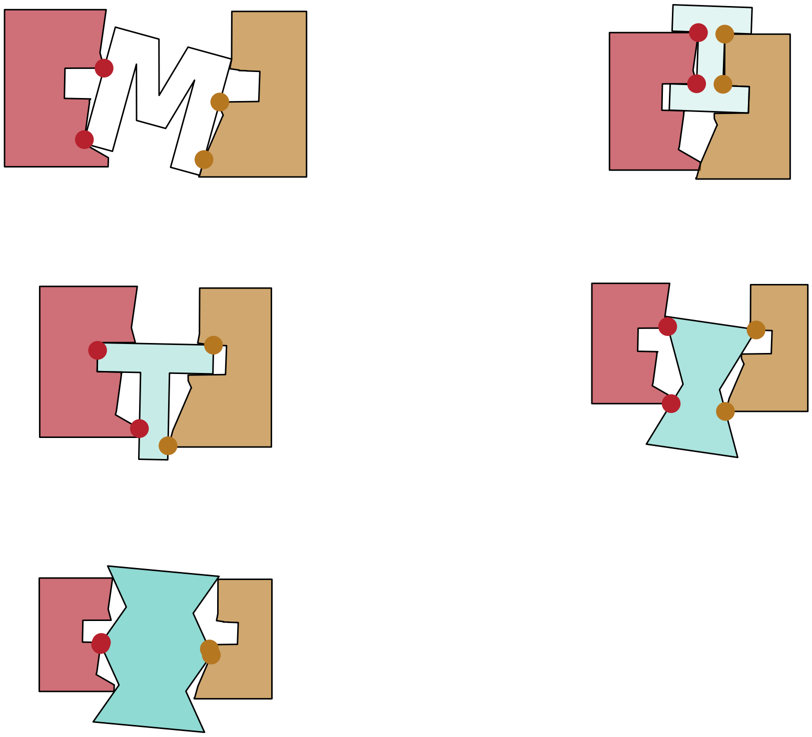


Figure 3-14: Best-cost compact, torque-robust gripper solution for complex shapes.

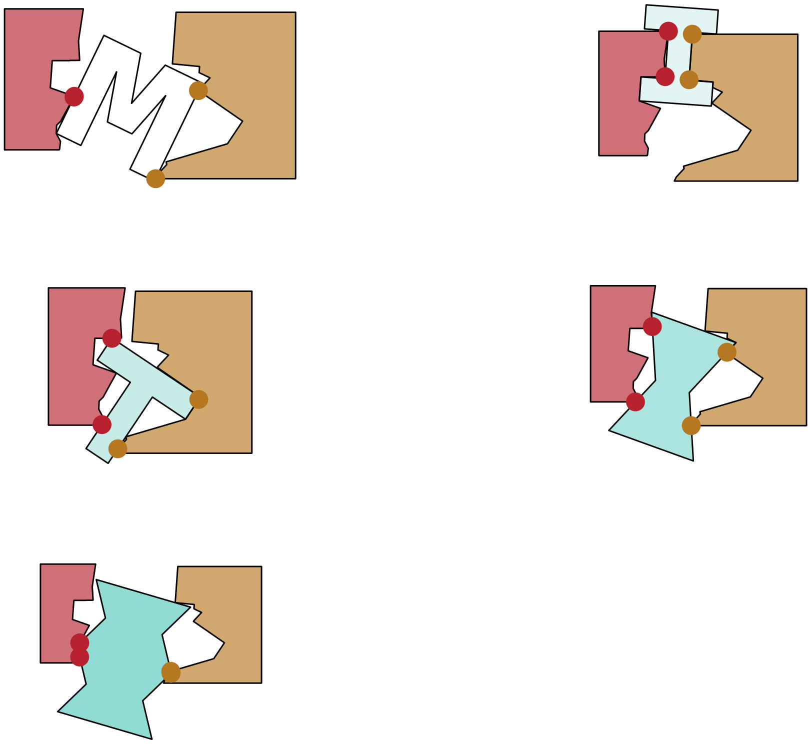


Figure 3-15: Second-best-cost compact, torque-robust gripper solution for complex shapes.

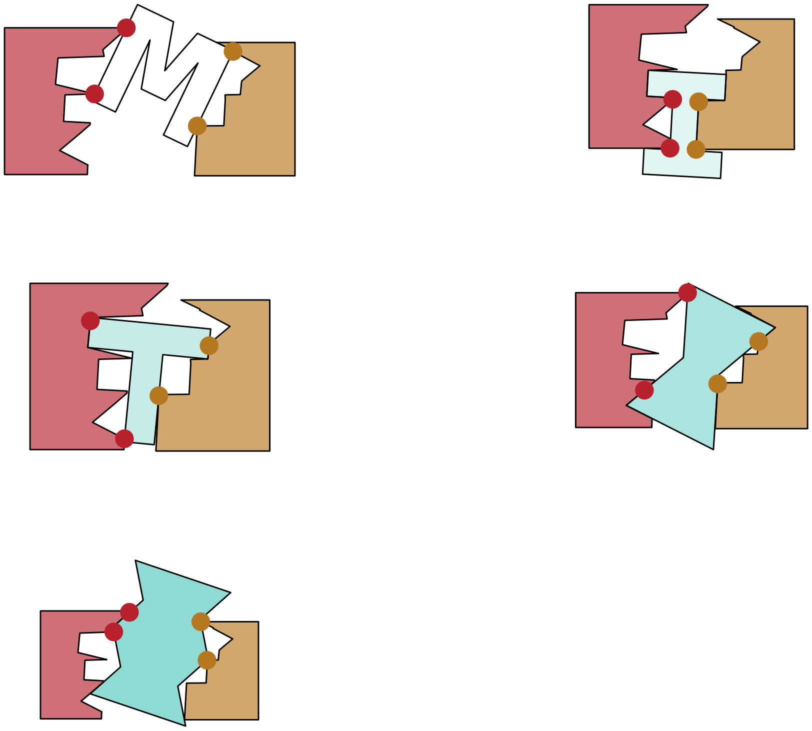


Figure 3-16: Third-best-cost compact, torque-robust gripper solution for complex shapes.

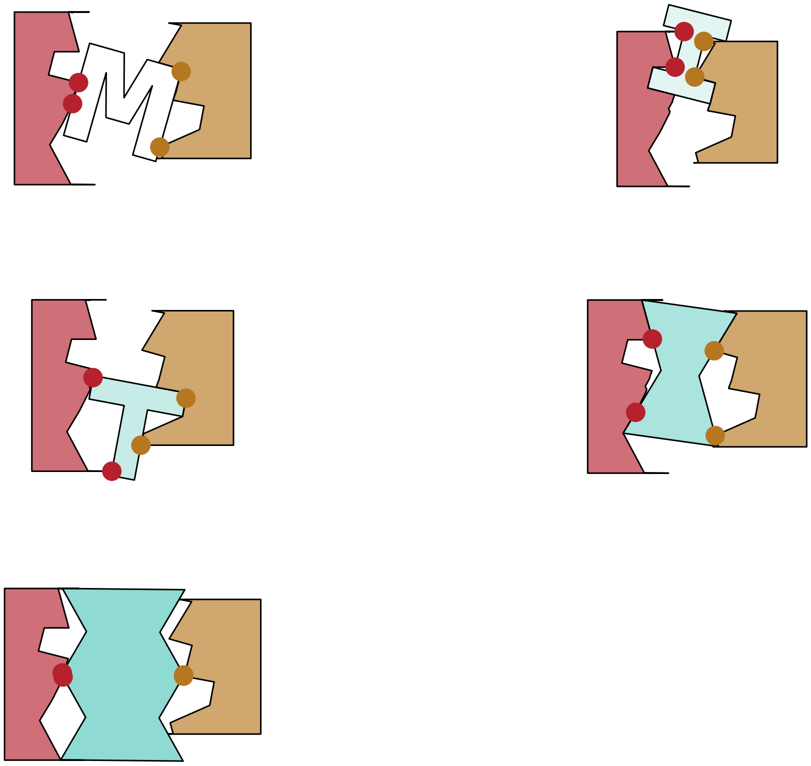


Figure 3-17: Fourth-best-cost compact, torque-robust gripper solution for complex shapes.

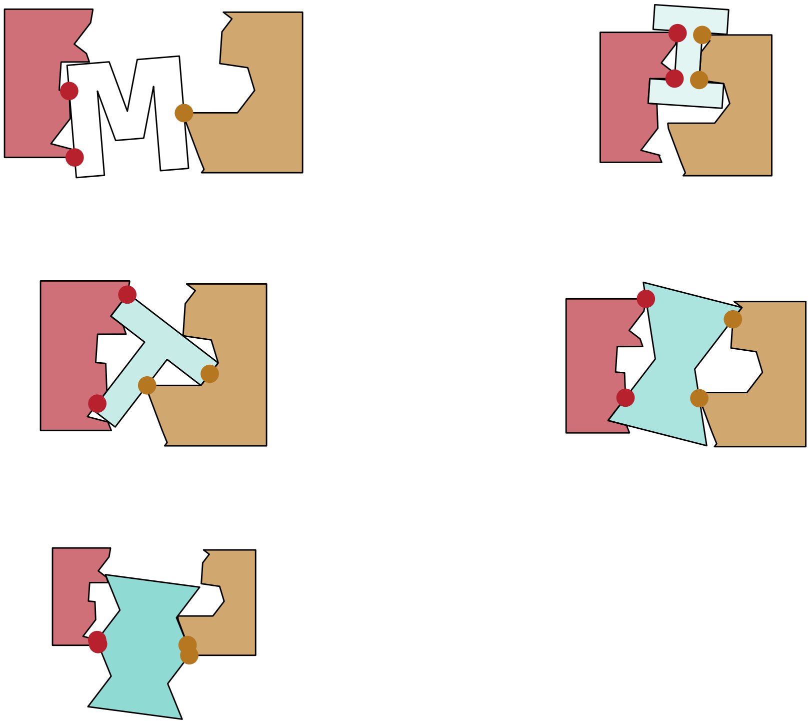


Figure 3-18: Fifth-best-cost compact, torque-robust gripper solution for complex shapes.

3.5.3 Modifications for preventing effector interpenetration

The proposed method has not prevented effector-effector interpenetration such as in Fig. 3-19. Thus far, the contact points themselves, in the effector frames, have been the only knowledge of the effector surfaces within the NLP optimization routine. However, as the points occupy no volume in the plane, it is not possible to check penetration between the two sets of points. To address this, each contact point can be assumed to be accompanied by some amount of material on the effector. For example, in the grasps defined by the contact points shown in Fig. 3-20 (the same solution as is illustrated in Fig. 3-19), without constructing the full effector jaw geometry, we can choose to assume that each contact is met by effector material within the trapezoids. We can then transform these trapezoids into the opposing effector's frame, sweep them along the linear path of relative motion of the two effectors, similarly to the process in Sec. 3.3, and add them to the composite shapes. The NLP solution used in Fig. 3-19 is visualized in the effector frames with these trapezoidal interpenetration keep-out shapes in Fig. 3-21, where the interpenetration near the bottom of the effector can be predicted by the fact that some contact points lie within some trapezoidal

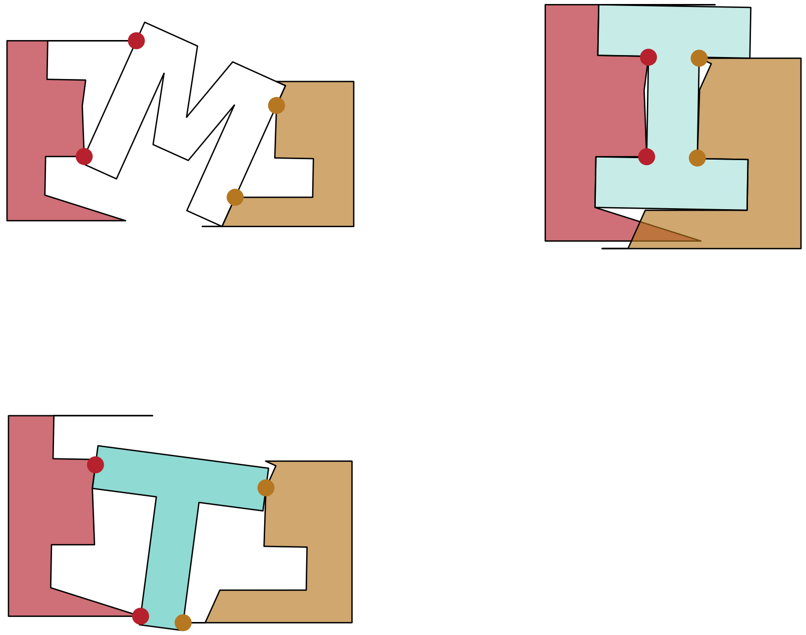


Figure 3-19: Grasp solution that leads to interpenetration between the two effector jaws.

keep-out shapes. In contrast, the solution visualized in Fig. 3-22, where the contact points lie outside all shapes in the effector frames, respects the new non-penetration constraint developed to prevent effector-effector interpenetration.

Upon solving the NLP such that contact points lie outside the composite shapes including the interpenetration keep-out shapes, the effector can be composed without causing effector-effector interpenetration by constructing features within these trapezoids. For example the rounded feature in Fig. 3-23 would be a reasonable choice because it is enveloped by the trapezoid. Repeating this feature once to meet each contact point in the feasible solution shown in Fig. 3-22 leads to the effector solution shown in Fig. 3-24, which indeed has no effector-effector interpenetration.

It would be more precise to enforce that the trapezoids do not intersect each other or the objects. The present formulation has simply enforced that the trapezoids do not intersect the contact points, as the non-penetration constraint is constructed to measure signed distance between shapes and points. However, this could be extended to measure distances between shapes, for example, by calculating the minimum distance between the sets. However, with polygonal keep-out shapes and objects, this constraint would suffer from even more issues with non-smoothness than the current SDF constraint does.

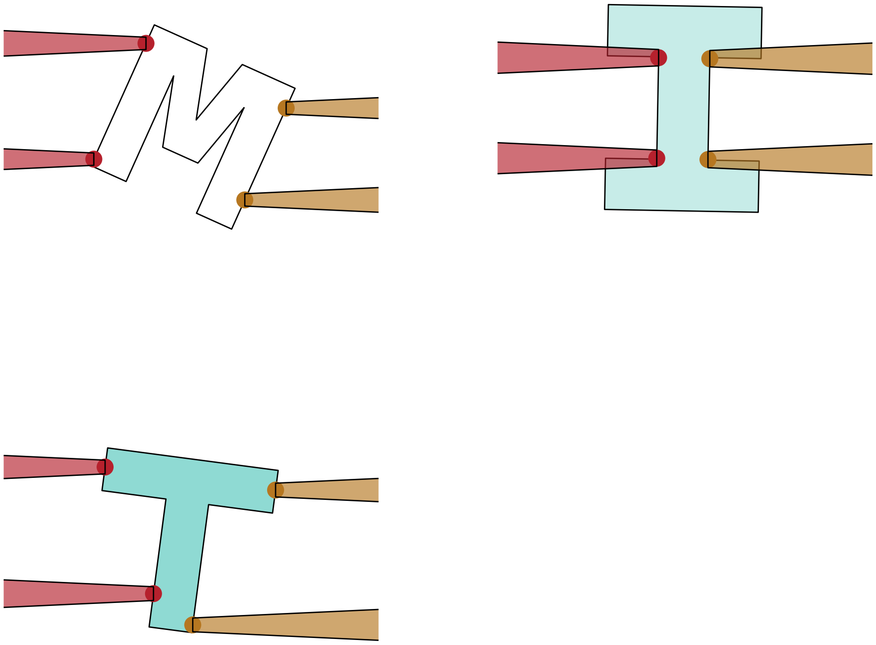
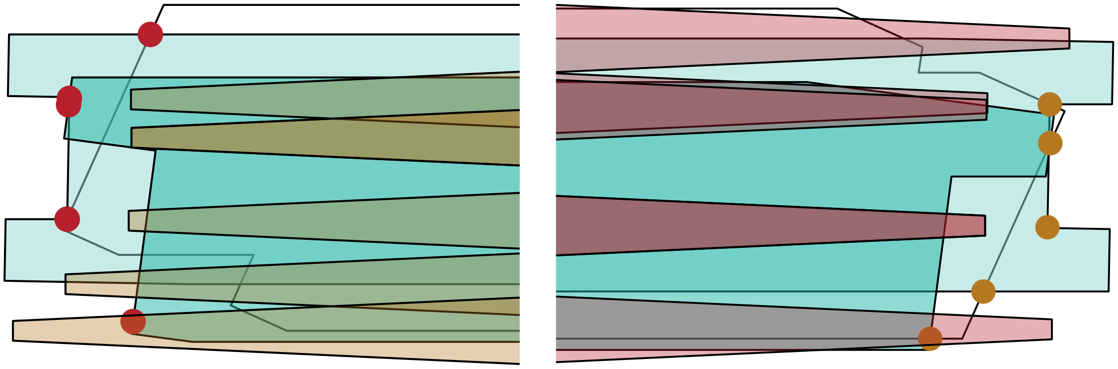


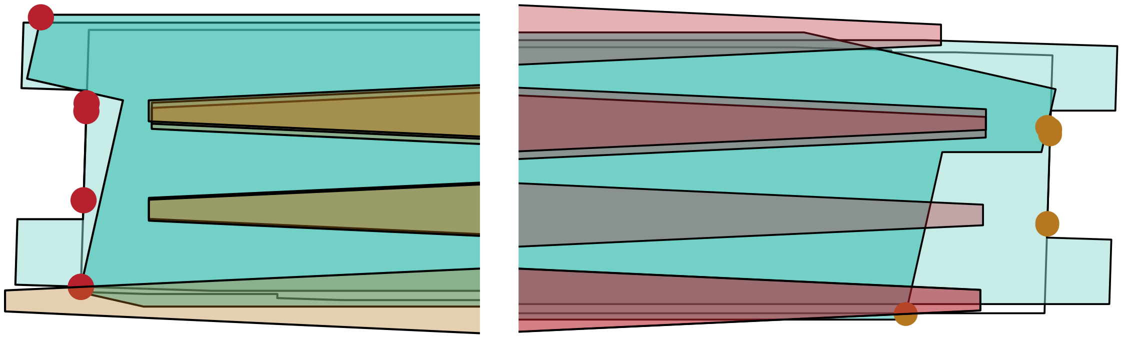
Figure 3-20: NLP solution that leads to interpenetration between the two effector jaws, with interpenetration keep-out shapes shown as red and yellow trapezoids.



(a)

(b)

Figure 3-21: Swept objects shown in the effector frames, $\tilde{\Psi}^{E_0}$ (a), $\tilde{\Psi}^{E_1}$ (b), for solution with effector interpenetration shown in Fig. 3-19. Yellow trapezoids in (a) are interpenetration keep-out shapes corresponding to contact points from effector 1, and red trapezoids in (b) are interpenetration keep-out shapes corresponding to contact points from effector 0. This solution leads to effector-effector interpenetration, which can be identified by the fact that the bottom contact point in each effector frame lies inside a trapezoidal interpenetration keep-out shape.



(a)

(b)

Figure 3-22: Swept objects shown in the effector frames, $\tilde{\Psi}^{E_0}$ (a), $\tilde{\Psi}^{E_1}$ (b), for solution without effector interpenetration shown in Fig. 3-24. Yellow trapezoids in (a) are interpenetration keep-out shapes corresponding to contact points from effector 1, and red trapezoids in (b) are interpenetration keep-out shapes corresponding to contact points from effector 0. Contact points are drawn as large circles for illustration, which may create the illusion that they penetrate shapes in this figure, but only the centers of these circles must be outside all shapes.

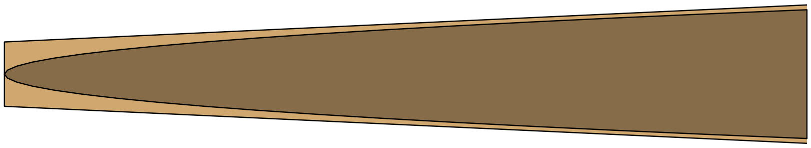


Figure 3-23: Proposed shape for effector feature (gray), bounded by interpenetration keep-out shape (yellow).

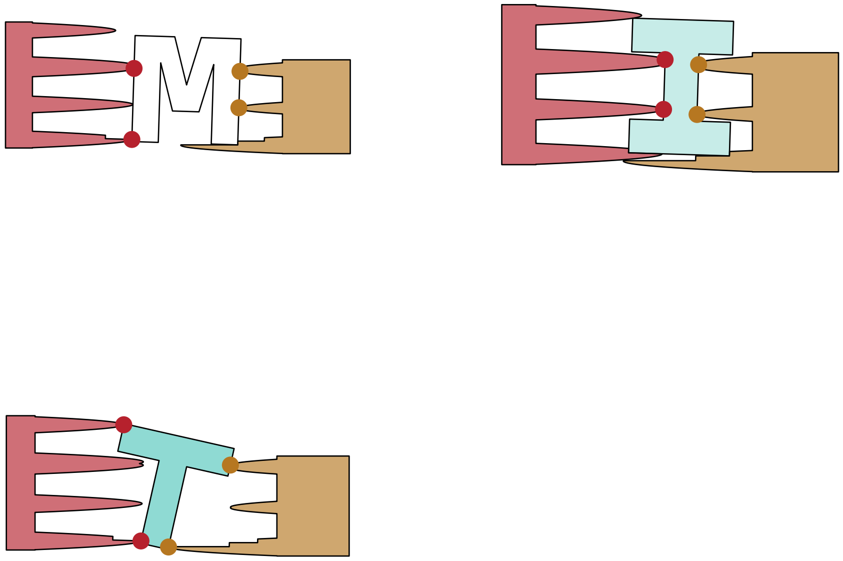


Figure 3-24: Grasp solution that does not lead to interpenetration between the two effector jaws.

Problem	μ	w	N_{iter}	T_s (hours)
Compact grippers – regular polygons	0.3	1	195	5
Torque robustness – regular polygons	0.3	1	20	0.5
Torque robustness – nonconvex	0.3	0.2	90	12

Table 3.1: Computational parameters for the example problems in Sec. 3.5. Here, N_{iter} is the number of NLP runs (each with a different randomized initial guess). T_s is the approximate duration of the entire run (including all NLP runs).

3.6 Parameters and computation

The number of optimization variables is linear in N_t. All presented problems are solved on a computer with Intel Core i7-10750H 2.60GHz CPU. As discussed in Sec. 3.1.2, the NLP is repeatedly rerun with different initial guesses. When to terminate this process is a decision that can be made by a user. Parameters used in the presented examples are shared in Table 3.1. NLPs are solved using MATLAB's `fmincon` [36].

Chapter 4

Conclusions

NLP formulations are presented for co-optimizing effector shape and motion for robotic tasks in two dimensions. Discretized continuous-time tasks with specified contact trajectories for rigid effectors are handled in Chap. 2, and the algorithms are extended to include internal effector DOFs, optimization over contact locations, and force balance in Chap. 3. Future work will leverage these extensions in two ways. First, it will incorporate contact trajectory optimization in the continuous problem, so that the full design space of all motions is simultaneously considered with shape. Second, it will leverage the ability to include internal DOFs in the continuous problem, adding the option to include additional joints.

Finally, this work may be generalized to three dimensions, where instead of designing one-dimensional effector curves, the framework can design two-dimensional effector surfaces. The non-penetration constraint extends to three dimensions, as SDFs can be computed for three-dimensional objects. In the continuity constraint, tangent vectors become tangent planes. Three-dimensional effectors with additional DOFs would be able to execute more interesting tasks and take on richer geometries.

4.1 Contact trajectory satisfaction

In Chap. 2, an NLP framework is proposed for co-optimizing shape and motion of rigid effectors, in particular for applications where sliding contact and detailed effector geometries are important. Instead of parameterizing the effector geometry, the framework represents the object and contact constraints in a frame that moves with the effector and impose that, in this frame, the contact points (a) do not penetrate the object and (b) lie on an integral of the contact tangents. This formulation yields solutions to several example problems, which validate that the framework can recover known solutions, optimize a variety of user-specified, application-specific cost functions, and find feasible solutions to complex problems that are difficult to solve with existing methods. This work also includes an experimental demonstration of a synthesized effector being used by a robotic arm.

4.2 Grasping

Chap. 3 provides a demonstration of an extension of the framework presented in Chap. 2 to include force balances and internal effector DOFs. As this work has focused less on extracting the effector geometry from the NLP solution, additional work in this area may improve the usefulness of the results. Secs. 4.2.1 to 4.2.3 discuss additional challenges and future directions for this work.

4.2.1 Optimizing over contact existence and edge correspondence

It has been assumed that the user inputs the correspondence between both contacts and effector jaws, and contacts and object edges, and thus also implicitly specifies the number of contacts. Relaxing any of these assumptions, and instead optimizing over these variables, would expand the optimization space, improving the generality of this approach. However, these factors are intrinsically discrete, and therefore would necessarily introduce integer variables to the program, transforming the problem into a mixed-integer nonlinear program (MINLP). While MINLPs are challenging to solve, they can be approached via methods such as branch-and-bound and convex relaxation [45]. The increased size and complexity of the optimization space introduced by these integer variables would lead to significantly longer solve times, which may be acceptable depending on the application. However, the problem could become intractable as the number of objects and number of contacts and edges per object grows.

4.2.2 Form closure

A form closure grasp is a grasp that restrains an object from undergoing any motion, independent of external wrenches [48]. Form closure grasps do not rely on frictional contact forces, but rather use contacts to geometrically prevent object motion. Thus, they can be desirable for ensuring that objects remain securely grasped under high or uncertain loads. As form closure is highly reliant on the geometry of the interactions between effector and object, optimizing form closure grasps may be a natural extension of the proposed method. For example, sets of contact placements for form closure grasps can be pre-computed [53], and these sets can be optimized over, within the proposed method to design corresponding effectors.

4.2.3 Robustness

Thus far, no notion of robustness has been discussed in this work. Consider the grasp of the M shape in Fig. 3-15. In particular, at the upper-right corner of the M, slight changes to the object geometry, the effector geometry, or the position of the effector relative to the object would prevent contact from being made as planned. Likewise, in the grasp of the T shape in Fig. 3-14, slight errors in the vertical positioning of the effector would cause the effector to fail to close around the shape. The presented framework could be improved by, for example, including a robustness metric, such as is developed by Kim et al. using a Monte Carlo approach [31].

A potential use of the presented framework is to design effectors robust to object shape variations. As the framework designs effectors that can grasp sets of objects, it can be used to design an effector for a set of variations of a single object geometry. In fact, if the set of objects approximates a continuum of variations over some parameters, the problem takes on some attributes of the continuous shape and motion problem from Chap. 2. The continuity constraint becomes once again relevant, and can be used to help construct the effector geometry from the NLP solution.

Appendix A

Signed distance function smoothness

SDFs are smooth at points not equidistant from ≥ 2 closest points on the boundary of the shape. Shown in Fig. A-1, this equidistant condition for SDF non-smoothness occurs for a circle at the center of the circle, for a square in four linear ridges inside the square, and for a star in five ridges outside the star and five (different) ridges inside the star.

Fig. A-2 shows how the screwdriver problem, first introduced in Chap. 1, is prone to having a region of non-smooth SDF. This sometimes causes problems in the gradient descent algorithm, particularly because there are contact points in this region and so the SDFs will be sampled there. Note that in this case, the "zig-zag" ridges in the composite shape are an artifact of the discretization of the motion – if the composite shape were generated via a continuous sweep of the shape, the corner shown in Fig. A-2a would trace a smooth curve.

The swept area can be more accurately approximated by taking the union of N approximate swept areas between each pair of consecutive timesteps. The kth approximate swept area (between timesteps k and $k+1$) is found by taking the union of effector-frame objects from these timesteps ($u[k] := \Psi^E[k] \cup \Psi^E[k+1], k = 0, \ldots, N_t - 1$), and removing from $u[k]$ vertices that (a) are not object vertices and (b) fall between instances of the same object vertex. For example, the internal corners of the "zig-zag" feature noted in Fig. A-2a would be removed, as seen in Fig. A-3, because they (a) are not vertices of the original object and (b) fall between instances of of the bottom-left vertex of the screwdriver shape, when traversing around the boundaries of the unions of consecutive Ψ^Es. This computation

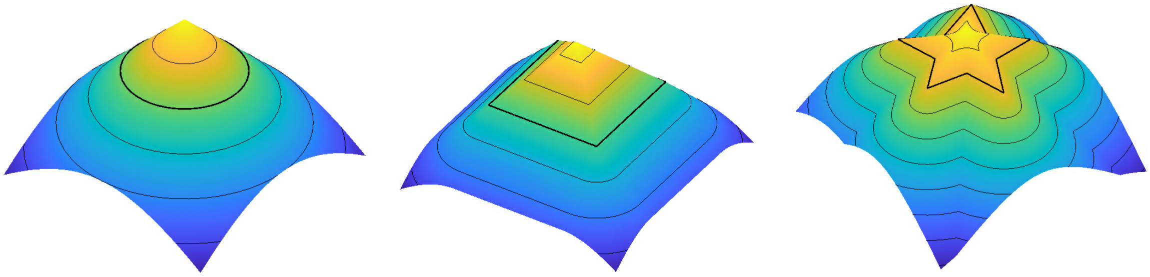
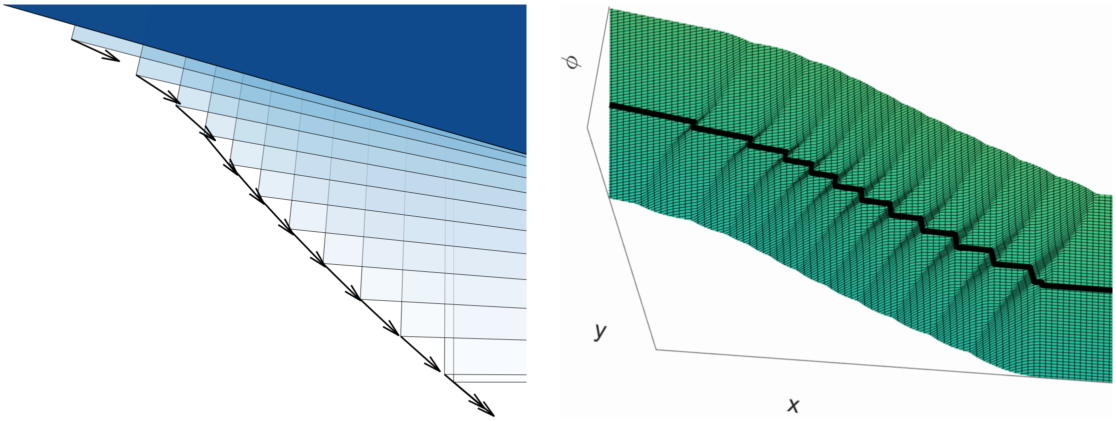


Figure A-1: SDFs of a circle, square, and star.



(a)

(b)

Figure A-2: (a) Corner zoom-in of the effector-frame representation of a solution to the screwdriver problem. (b) SDF of this shape, near this area. The thick black curve is the zero level set (shape boundary).

is more expensive than the original discrete union method, but offers some benefit to the optimizer.

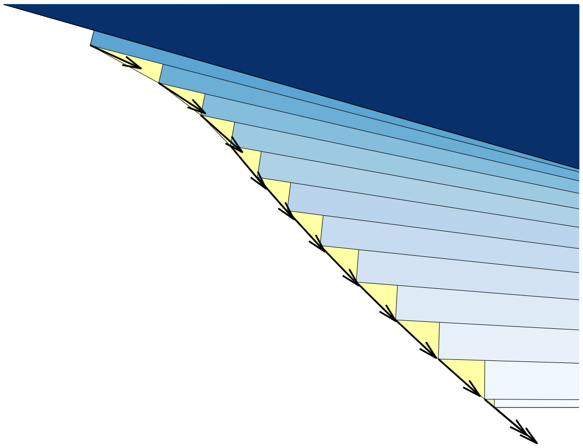


Figure A-3: Overlay of multiple frames of a shape moving through a trajectory, laid on top of the approximated swept area (yellow).

Bibliography

[1] Program Solicitation DARPA-PS-16-01 for Robotic Servicing of Geosynchronous Satellites (RSGS). `https://acquisitioninnovation.darpa.mil/docs/DARPA%20OT%20Programs/RSGS.pdf`, May 2016. (Accessed on: Oct. 16, 2021). 18

[2] NASA EDGE: CIRAS Lead Systems Engineer. `https://www.youtube.com/watch?v=qTNv9exASFA`, July 2018. (Accessed on: Oct. 16, 2021). 18

[3] Bernardo Aceituno-Cabezas and Alberto Rodriguez. A global quasi-dynamic model for contact-trajectory optimization. In *Robotics: Science and Systems*, 2020. 19

[4] Ferran Alet, Maria Bauza, Adarsh K Jeewajee, Max Thomsen, Alberto Rodriguez, Leslie Pack Kaelbling, and Tomás Lozano-Pérez. Robotic gripper design with evolutionary strategies and graph element networks. In *NeurIPS Workshop on Machine Learning for Engineering Modeling, Simulation, and Design*, 2020. 23

[5] R Ambrose, IAD Nesnas, F Chandler, BD Allen, T Fong, L Matthies, and R Mueller. NASA technology roadmaps: TA 4: Robotics and autonomous systems. *NASA, Washington DC*, 2015. 18

[6] Kendall Atkinson. *An Introduction to Numerical Analysis, 2nd Edition*. John Wiley and Sons, Inc., 1991. 29

[7] Cenk Baykal, Chris Bowen, and Ron Alterovitz. Asymptotically optimal kinematic design of robots using motion planning. *Autonomous Robots*, 43(2):345–357, 2019. 22

[8] Wendel K Belvin, William R Doggett, Judith J Watson, John T Dorsey, Jay E Warren, Thomas C Jones, Erik E Komendera, Troy Mann, and Lynn M Bowman. In-space structural assembly: Applications and technology. In *AIAA Spacecraft Structures Conference*, page 2163, 2016. 17

[9] John T Betts. *Practical methods for optimal control and estimation using nonlinear programming*. SIAM, 2010. 29

[10] Russell G Brown and Randy C Brost. A 3-D modular gripper design tool. *IEEE Transactions on Robotics and Automation*, 15(1):174–186, 1999. 23

[11] Tianjian Chen, Zhanpeng He, and Matei Ciocarlie. Hardware as policy: Mechanical and computational co-optimization using deep reinforcement learning. In *Conference on Robot Learning*, 2020. 22

[12] Xianyi Cheng, Eric Huang, Yifan Hou, and Matthew T Mason. Contact mode guided sampling-based planning for quasistatic dexterous manipulation in 2D. In *IEEE International Conference on Robotics & Automation*, pages 6520–6526, 2021. 19

[13] Krishnamanaswi M Digumarti, Christian Gehring, Stelian Coros, J Hwangbo, and Roland Siegwart. Concurrent optimization of mechanical design and locomotion control of a legged robot. In *Mobile Service Robotics*, pages 315–323. World Scientific, 2014. 22

[14] William R Doggett, John Dorsey, John Teter, David Paddock, Thomas Jones, Erik E Komendera, Lynn Bowman, Chuck Taylor, and Martin Mikulas. Persistent assets in zero-g and on planetary surfaces: Enabled by modular technology and robotic operations. In *AIAA SPACE Forum*, page 5305, 2018. 17

[15] William R Doggett, John T Dorsey, Concepts Branch, and David S Kang. State of the Profession Considerations: NASA Langley Research Center Capabilities and Technologies for Large Space Structures, In-Space Assembly and Modular Persistent Assets. `http://surveygizmoresponseuploads.s3.amazonaws.com/fileuploads/623127/5043187/194-e16689534e8d64b93af3781b11ca5b1e_Langley_ISA_APC_White_Paper_V9.pdf`, 2019. (Accessed on: May. 17, 2022). 18

[16] Neel Doshi, Francois R Hogan, and Alberto Rodriguez. Hybrid differential dynamic programming for planar manipulation primitives. In *IEEE International Conference on Robotics & Automation*, pages 6759–6765, 2020. 19

[17] Pierre-Alain Fayolle. Distance to set operations in constructive modeling of solids. Technical report, The University of Aizu, Computer Graphics Laboratory, 2009. 40

[18] Thomas Geijtenbeek, Michiel Van De Panne, and A Frank Van Der Stappen. Flexible muscle-based locomotion for bipedal creatures. *ACM Transactions on Graphics*, 32(6):1–11, 2013. 22

[19] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47:99–131, 2005. 33

[20] Philip E. Gill, Walter Murray, Michael A. Saunders, and Elizabeth Wong. User's guide for SNOPT 7.7: Software for large-scale nonlinear programming. Center for Computational Mathematics Report CCoM 18-1, Department of Mathematics, University of California, San Diego, La Jolla, CA, 2018. 33

[21] David Ha. Reinforcement learning for improving agent design. *Artificial Life*, 25(4):352–365, 2019. 22

[22] Huy Ha, Shubham Agrawal, and Shuran Song. Fit2Form: 3D Generative Model for Robot Gripper Form Design. In *Conference on Robot Learning*, 2020. 23

[23] Sehoon Ha, Stelian Coros, Alexander Alspach, Joohyung Kim, and Katsu Yamane. Computational co-optimization of design parameters and motion trajectories for robotic systems. *International Journal of Robotics Research*, 37(13-14):1521–1536, 2018. 22

[24] Maximilian Haas-Heger. *Grasp Stability Analysis with Passive Reactions*. PhD thesis, Columbia University, 2021. 45

[25] Maximilian Haas-Heger, Christos Papadimitriou, Mihalis Yannakakis, Garud Iyengar, and Matei Ciocarlie. Passive static equilibrium with frictional contacts and application to grasp stability analysis. In *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, June 2018. 44, 45

[26] Christopher Hazard, Nancy Pollard, and Stelian Coros. Automated design of manipulators for in-hand tasks. In *IEEE-RAS International Conference on Humanoid Robots*, pages 1–8, 2018. 22

[27] Robert Hoyt, Jesse Cushing, Greg Jimmerson, Jeffrey Slostad, Robert Dyer, and Steve Alvarado. Spiderfab: Process for on-orbit construction of kilometer-scale apertures. *Final Report of the NASA NNX12AR13G Contract*, 2013. 18

[28] Alec Jacobson, Ilya Baran, Jovan Popovic, and Olga Sorkine. Bounded biharmonic weights for real-time deformation. *ACM Transactions on Graphics*, 30(4):78, 2011. 22

[29] Kento Kawaharazuka, Toru Ogawa, and Cota Nabeshima. Tool shape optimization through backpropagation of neural network. In *IEEE International Conference on Intelligent Robots and Systems*, pages 8387–8393, 2020. 23

[30] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, 1995. 43

[31] Junggon Kim, Kunihiro Iwamoto, James J Kuffner, Yasuhiro Ota, and Nancy S Pollard. Physically based grasp quality evaluation under pose uncertainty. *IEEE Transactions on Robotics*, 29(6):1424–1439, 2013. 69

[32] Scott Kirkpatrick, C Daniel Gelatt Jr, and Mario P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. 43

[33] Gilwoo Lee, Tomás Lozano-Pérez, and Leslie Pack Kaelbling. Hierarchical planning for multi-contact non-prehensile manipulation. In *IEEE International Conference on Intelligent Robots and Systems*, pages 264–271, 2015. 19

[34] Kevin Sebastian Luck, Heni Ben Amor, and Roberto Calandra. Data-efficient co-adaptation of morphology and behaviour with deep reinforcement learning. In *Conference on Robot Learning*, pages 854–869. PMLR, 2020. 22

[35] Yusuke Maeda, Tomohisa Nakamura, and Tamio Arai. Motion planning of robot fingertips for graspless manipulation. In *IEEE International Conference on Robotics & Automation*, volume 3, pages 2951–2956, 2004. 19

[36] Matlab optimization toolbox, 2020. The MathWorks, Natick, MA, USA. 33, 66

[37] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 137–144, 2012. 19

[38] Anusha Nagabandi, Kurt Konolige, Sergey Levine, and Vikash Kumar. Deep dynamics models for learning dexterous manipulation. In *Conference on Robot Learning*, pages 1101–1112. PMLR, 2020. 19

[39] Tønnes F Nygaard, Charles P Martin, Eivind Samuelsen, Jim Torresen, and Kyrre Glette. Real-world evolution adapts robot morphology and control to hardware limitations. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 125–132, 2018. 22

[40] Chandana Paul and Josh C Bongard. The road less travelled: Morphology in the optimization of biped robot locomotion. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, volume 1, pages 226–232, 2001. 22

[41] Michael Posa, Cecilia Cantu, and Russ Tedrake. A direct method for trajectory optimization of rigid bodies through contact. *International Journal of Robotics Research*, 33(1):69–81, 2014. 19

[42] Máximo Alejandro Roa Garzon, Korbinian Nottensteiner, Armin Wedler, and Gerhard Grunwald. Robotic technologies for in-space assembly operations. In *14th Symposium on Advanced Space Technologies in Robotics and Automation*, 2017. 18

[43] Alberto Rodriguez and Matthew T Mason. Grasp invariance. *International Journal of Robotics Research*, 31(2):236–248, 2012. 19, 27, 28, 33, 36

[44] Alberto Rodriguez and Matthew T Mason. Effector form design for 1DOF planar actuation. In *IEEE International Conference on Robotics & Automation*, pages 349–356, 2013. 19, 28, 36

[45] Nikolaos V Sahinidis. Mixed-integer nonlinear programming 2018. *Optimization and Engineering*, 20(2):301–306, 2019. 68

[46] Charles Schaff, David Yunis, Ayan Chakrabarti, and Matthew R Walter. Jointly learning to construct and control agents using deep reinforcement learning. In *IEEE International Conference on Robotics & Automation*, pages 9798–9805, 2019. 22

[47] Lukas Christoffer Malte Wiuf Schwartz, Adam Wolniakowski, Andrzej Werner, Lars-Peter Ellekilde, and Norbert Krüger. Designing fingers in simulation based on imprints. In *International Conference on Simulation and Modeling Methodologies, Technologies and Applications*, pages 304–313, 2017. 23

[48] Bruno Siciliano, Oussama Khatib, and Torsten Kröger. *Springer handbook of robotics*, volume 200. Springer, 2008. 45, 69

[49] Brook Sullivan, David Barnhart, Lisa Hill, Paul Oppenheimer, Bryan L Benedict, Gerrit Van Ommering, Laurie Chappell, John Ratti, and Peter Will. DARPA phoenix payload orbital delivery system (PODs): "FedEx to GEO". In *AIAA SPACE Forum*, page 5484, 2013. 17

[50] Al Tadros and John Lymer. NASA Restore-L Program. `https://www.spacefoundation.org/wp-content/uploads/2019/07/Presentation-Tadros-Al-NASA-Restore-L-Program.pdf`, 2019. (Accessed on: May. 17, 2022). 18

[51] Orion Taylor and Alberto Rodriguez. Optimal shape and motion planning for dynamic planar manipulation. *Autonomous Robots*, 43(2):327–344, 2019. 22

[52] Marc A Toussaint, Kelsey Rebecca Allen, Kevin A Smith, and Joshua B Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. In *Robotics: Science and Systems*, 2018. 19

[53] A Frank van der Stappen, Chantal Wentink, and Mark H Overmars. Computing immobilizing grasps of polygonal parts. *International Journal of Robotics Research*, 19(5):467–479, 2000. 69

[54] Kevin Wampler and Zoran Popović. Optimal gait and form for animal locomotion. *ACM Transactions on Graphics*, 28(3):1–8, 2009. 22

[55] Adam Wolniakowski, Jimmy A Jorgensen, Konstantsin Miatliuk, Henrik Gordon Petersen, and Norbert Kruger. Task and context sensitive optimization of gripper design using dynamic grasp simulation. In *International Conference on Methods and Models in Automation and Robotics*, pages 29–34. IEEE, 2015. 23

[56] J Zachary Woodruff and Kevin M Lynch. Planning and control for dynamic, nonprehensile, and hybrid manipulation tasks. In *IEEE International Conference on Robotics & Automation*, pages 4066–4073, 2017. 19

[57] Jie Xu, Tao Chen, Lara Zlokapa, Michael Foshey, Wojciech Matusik, Shinjiro Sueda, and Pulkit Agrawal. An end-to-end differentiable framework for contact-aware robot design. In *Robotics: Science and Systems*, 2021. 22

[58] Connor L Yako, Shenli Yuan, and J Kenneth Salisbury. Designing underactuated graspers with dynamically variable geometry using potential energy map based analysis. *arXiv preprint arXiv:2203.07456*, 2022. 23