Modeling, Simulation, and Design of Self-Assembling Space Systems:

Accurate Collision Detection, Robust Time Integration, and Optimal Control

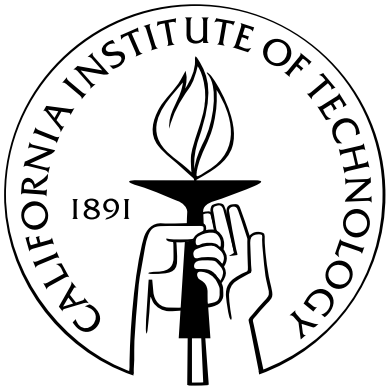
Thesis by

Gwendolyn B. Johnson

In Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy



California Institute of Technology

Pasadena, California

2013

(Defended August 27, 2012)

Acknowledgements

I would like to begin by thanking the members of my thesis committee–Drs. Michael Ortiz, Sigrid Leyendecker, Sergio Pellegrino, and Ravi Ravichandran–for their guidance, feedback, and advice during my time at Caltech. To begin, I am indebted my advisor Dr. Michael Ortiz for his enthusiasm and encouragement to pursue topics that I found truly interesting and exciting, and for providing the opportunity to teach and travel as part of my time at Caltech. Michael, you've always received A+ ratings on Caltech's annual (confidential until this point) mental health surveys. In addition to being a close collaborator and outstanding proofreader, Dr. Sigrid Leyendecker and her research group have been gracious hosts for two summers in Kaiserslautern and Erlangen. The main direct application of my work thus far is related to the AAReST mission, a project funded by the Keck Institute for Space Studies jointly with Caltech and JPL and supervised by Dr. Sergio Pellegrino, who has provided valuable feedback on the more practical aspects of my work. Finally, my thanks to Dr. Ravi Ravichandran for allowing me to take up as much of his time as needed for both technical and professional advice.

The support staff for my committee members, Lydia Suarez, Marta Kahl, Cheryl Gause, and Beate Hegen should also be acknowledged for the skill it takes to keep people, machines, and plans running smoothly.

In addition, the members of the KISS (or AAReST...or whatever the new acronym becomes) group John Baker, Dr. James Breckinridge, Dr. Chiara Daraio, Terry Gdoutos, Dr. Marin Kobilarov, Dr. Namiko Yamamoto, Keith Patterson, and John Steeves have provided useful insights and keeping up on their work has been the 'spice' in my academic life at Caltech. In particular, I would like to thank Marin for his outstanding work on building the experimental testbed for the AAReST satellites and his collaboration on the feedback controller design for the system–and his tolerance of my soldering technique. Dr. Tim Sullivan, has also provided useful feedback and instruction on the analysis contained in this work, and has a gift for getting others excited about mathematics.

Although it can feel lonely at times, the process of earning an advanced degree would not be possible

without family and friends. To my parents, who inspired and encouraged my interest in engineering and hopefully didn't regret it after they paid for my undergraduate education at M.I.T., thank you. I'll get to work on your massive retirement fund ASAP if you'll still let me do laundry at your house. To my big brother Evan and my friends from Phoenix–some of which I've known for the better part of 20 years–and my friends from M.I.T., thank you for accepting my quirks and only judging me mildly for being a nerd. Last, to my Caltech friends and in particular Sebastian and my roommates of more than three years–two legged and four–this has been a lot more fun than I ever thought it could be. May be we're not working hard enough.

Finally, this work would not have been possible without generous support from the Keck Institute for Space Studies.

Abstract

Motivated by issues inherent in modeling and designing self-assembling systems (e.g. multiple collisions, collisions between non-smooth bodies, clumping and jamming behaviors, etc.), the goal of this thesis is to develop robust numerical tools that enable efficient and accurate direct simulation of self assembling systems and the application of optimal control methods to this type of system. The systems will be alternately modeled using linear finite elements, rigid bodies, or chains of rigid bodies. To this end, this work begins with development of a linear programming based collision detection algorithm for general convex polyhedral bodies. The resulting linear program has several features which render it extremely useful in determining the force system at the time of contact in numerical collision integrators. With robust collision detection in hand, three related numerical integration methods for dynamics with collisions are treated; a direct potential-based approach, and exact collision integrator in a discrete variational setting, and a decomposition-based algorithm, again in the discrete variational setting. Finally, several control problems are treated in the Discrete Mechanics and Optimal Control–Constrained (DMOCC) framework in which collisions between non-smooth bodies either need to be avoided or explicitly included in the optimal control problem. A globally stable feedback controller and a family of trajectories for spacecraft docking are also developed and tested with an accurate representation of an optimized CubeSat docking system.

Contents

1 Introduction

1.1 Background

A self-assembling system is a system in which a collection of bodies (e.g., particles, robots) arrange themselves into a larger coherent structure which is typically endowed with different or augmented functionality. The ideas behind and motivation to build many self-assembling systems come from the widespread existence and apparent efficiency of self-assembling systems found in nature like virus capsids, membranes, and crystal growth (c. f. Hormoz and Brenner [35], Klavins [42], Whitesides and Gryzbowski [84]), and man made systems of this type are an active area of research from the nano to macro scales.

On the nano- and micro-scales, there is active research in optimizing the assembly of devices with applications in bioengineering, medicine, solar cells, and microdevices (c. f. Pankavich et al. [69], Peet et al. [71], Solomon et al. [79]). On larger scales, autonomous modular robotic systems and autonomously reconfigurable structures have also been an alluring topic for several years. Many authors have made progress in grammars for robotic communication (see Murata et al. [62, 63], Napp and Klavins [64]) which make use of graph grammars* so that a pre-programmed final structure can be built from what are typically identical modules. Probabilistic control laws for stochastic self-assembling systems, e. g. for large swarms of miniature robots, have also been investigated by Correll and Martinoli [18] and Tolley and Lipson [82]. What the works cited here as background have in common is the development of control laws, communication systems, and the investigation of the kinetics and kinematics of self-assembling systems in simulation. Many authors build small experiments to test their analysis but the *dynamics* of the proposed assembly process go unmodelled in computational investigations, which limits the scale on which these systems can be tested and characterized prior to implementation.

The use of self-assembling systems for space applications follows naturally from challenges inherent in

*Joyner et al. [37] is a good reference for this topic.

space exploration and space aperture construction, i.e. that the size of a spacecraft or space aperture is limited by current launch capabilities, and it is likely that humans will not be able to assist directly in the assembly of larger apertures. This type of space system has become increasingly attractive and feasible in recent years due to the development of several relatively low-cost 'micro'- and 'nano'-spacecraft platforms[†] and the expanding expertise in communications, guidance, navigation, and control for small satellites. Woellert et al. [88] provides a useful review of the popular CubeSat platform and Kenyon et al. [40] describes one of the most advanced low-cost small satellite missions to date using this platform. Several other groups have focused on developing specialized systems like the FIMER robots of Everist et al. [23] and the flux-pinned magnet-superconductor pair of Shoer and Peck [78] to facilitate the assembly of large space structures.

With self-assembling space systems well on their way to becoming a reality, the need for robust simulations of the dynamics of the assembly process to both design and test that process is apparent, and motivates the work contained in this thesis.

1.2 Motivation

The work contained in this thesis is largely motivated by the desire to understand and design systems capable of either actively or passively building themselves from the bottom up, with a focus on the application of such systems to space apertures. This focus, which represents a paradigm shift in the construction of large space telescopes, stems from a collaborative technical development grant from the Keck Institute for Space Studies (KISS) that aims to demonstrate the feasibility and practicality of various components of a modular telescope system, which will be demonstrated on the AAReST (Autonomous Assembly and Reconfiguration of a Space Telescope) mission (see Figure 1.1).

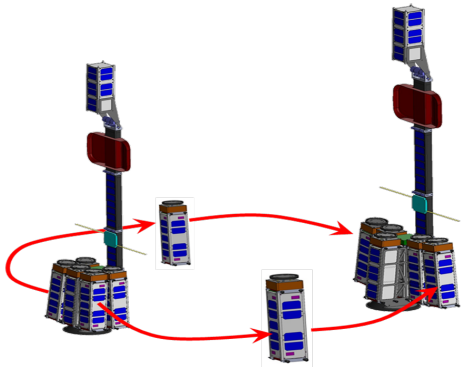


Figure 1.1: Concept for AAReST mission reconfigurable space telescope.

The problem of accurately and robustly modeling the dynamics of a self-assembling system presents

[†]That is, spacecrafts with dimensions on the order of tens of centimeters, as opposed to meters.

several challenges which, while also present in other types of systems, play a predominant role in the dynamics of the systems of interest. For example, the dynamics to be modeled include collisions, sustained contact (i.e. clumping or jamming), and complicated multiple-contact scenarios between stiff or rigid bodies. A general model also needs to account for non-smooth non-convex bodies and strong attractive forces at contact interfaces between bodies.

In addition to the problem of modeling the dynamics, the design and certification problems for this type of system present novel challenges in their own right in the areas of robust and optimal control, and model-based system design and certification. These problems take on several flavors for active and passive systems. For passive systems, one can think of controlling the dynamics to affect a desired final configuration via the design (geometry, inertial properties, potentials) of the modules themselves, subject to uncertainty. For active systems, control algorithms must be formulated to avoid contact between non-smooth bodies or to account for the fact that contact interactions will play a role in the dynamics. For computational studies of passive systems, an accurate description of the mechanical system including contact interactions is essential. In the case of active systems, a method of controller design that explicitly accounts for the non-smooth nature of the system is required.

Before moving on to a more detailed overview of how these challenges are met, it is worth noting that they are not unique to self-assembling systems. However, the extent to which they are present motivates the need for new tools to model, design, and better understand self-assembling systems, with the added benefit that, once developed, these tools become directly applicable to related problems.

1.3 Overview

This thesis is organized as follows. In Chapter 2, a novel approach to robust fine-scale collision detection for non-smooth polyhedral bodies is presented. This approach, which is called the Supporting Separating Hyperplane algorithm (or simply the SSH algorithm), is based on fundamental separation theorems for convex sets. It is shown that for polyhedral sets, the SSH algorithm may be evaluated as a linear program (the SSH LP), and that this linear program is always feasible and always subdifferentiable with respect to the configuration variables, which define the constraint set of the linear program. This is true regardless of whether the program is primal degenerate, dual degenerate, or both. The subgradient of the SSH linear program always lies in the normal cone of the closest admissible configuration to an inadmissible contact configuration. In particular if a contact surface exists, the subgradient of the SSH linear program is orthogonal to the contact surface, as

required of contact reactions. This property of the algorithm is particularly important in modeling stiff systems, rigid bodies, and tightly packed or jammed systems.

In Chapter 3, the application of the SSH LP to various dynamics formulations is presented. These formulations fall into three sub-categories: penalty methods, exact methods, and conserving approximations. In the final category, a parameter-free explicit contact algorithm for rigid body dynamics is developed for smooth or non-smooth geometries within the framework of a constrained variational integrator. This algorithm extends the decomposition contact response (DCR) algorithm for finite element dynamics of Cirak and West [15], and can be considered an explicit approximation to the fully implicit (exact) variational approach of Fetecau et al. [25]. For non-smooth bodies, collision detection and accurate momentum updates are enabled by the supporting separating hyperplane SSH algorithm.

With robust collision detection and simulations methods in hand, in Chapter 4 these concepts are combined and incorporated into optimal control problems with both collision avoidance and planned contacts between non-smooth bodies. This chapter is divided into two main sections. In the first, the structure preserving constrained optimal control methodology for discrete mechanical systems (DMOCC) introduced in Leyendecker et al. [52] is extended to include collision avoidance and planned contacts. The latter goal is accomplished without over-constraining the problem by allowing the physical contact time(s) and configuration(s) to vary in the course of the optimization. The final sections of both Chapters 3 and 4 summarize the application of the methods developed therein to real-world design and control problems of the AAReST mission.

As mentioned in the motivation for this work, the problems inherent in modeling and designing self-assembling systems are not necessarily unique to this type of system; it follows that the tools developed in this thesis can naturally be extended to enrich other fields. To this end, future research directions related to and inspired by the work presented in this thesis are presented in Chapter 5 alongside a summary of the essential achievements and results of the present work.

2 Non-Smooth Collision Detection

2.1 Introduction

In this chapter, a contact detection algorithm for non-smooth convex bodies is developed. The proposed contact detection algorithm can be concisely described as a supporting separating hyperplane (SSH) test for interpenetration, and is based on standard separation theorems for compact convex sets. The test is developed in detail for polyhedral bodies, where the SSH test can be effectively reformulated as a linear programming problem–the SSH LP. It is further shown that the subgradient of the SSH LP can be readily evaluated and–as will be discussed in detail in future chapters–that the subgradient supplies the force system at the time of contact.

2.2 Background and Related Work

2.2.1 Previous Work

A large body of literature exists on the efficient detection of collisions, driven in large part by advances in computational geometry, computer graphics and robotics (see Akgunduz et al. [2], Aliyu and Al-Sultan [3], Chakraborty et al. [13], Chung and Wang [14], Cohen et al. [17], Dobkin and Kirkpatrick [20], Ericson [22], Gilbert et al. [29], Gilbert and Hong [30], Gottschalk et al. [33], Li et al. [54], Lin et al. [56], Tang et al. [81]). While several advances have taken place in the past several years, the 2001 review by Jimenez et al. [36] and the 2005 book Ericson [22] effectively summarize the essential state of the art. The development of the SSH algorithm follows the same track as much of this literature, which develops collision detection algorithms for use with convex polyhedra and for solid models that have been discretized by polyhedra. However, the collision detection algorithm presented here is general enough to extend to an interpenetration test between any two smooth or non-smooth convex bodies, but the linear programming solution methodology is not, in general,

extensible to these situations.

The most popular software packages make use of hierarchical volume bounding to organize oriented bounding bodies (OBB's) and axis-aligned bounding bodies (AABB's) into rapidly searchable data structures (Octrees, K-D trees etc.) [17, 22, 33]. Intended for discretized surfaces, these algorithms can quickly compute candidate areas for contact, and refine those areas to determine which simplices are actually involved in a collision. However, when the objects get close enough for the bounding volumes to suggest that contact might have taken place, a finer detection test must be used to conclusively declare that a collision has taken place. Another option that works on the coarse and fine levels, proposed by Chung and Wang, detects collision based on the existence (or not) of a separating vector [14]. While the present interpenetration function is universal and robust, in that it can be evaluated for any pair of convex bodies, and is certainly capable of serving as a coarser scale collision detection test, one would rather suggest its use as a final test in conjunction with one of the coarser tests referenced here.

The interpenetration function here can be seen as an alternative to the heuristic search for a separating vector developed by Chung and Wang [14], and also as a an alternative to other linear programming approaches such as those proposed by Akgunduz and others [2] and Aliyu and Al-Sultan [3], to which Seidel [77] made key contributions. The advantage of the proposed linear programming approach to collision detection is that it provides extremely useful additional information for physics-based dynamics simulations and closest-point projection (CPP) operations. In the present work, a well known path to collision detection and undertaken in the form of a search for a separating vector. However, this search for this vector si conducted in an optimal way so that it always lies in the normal cone of the closest admissible configuration to an inadmissible contact configuration and respects any symmetries present in the geometry of the contact configuration. Furthermore, unlike previously proposed methods, the present approach ensures that the subgradient of the SSH linear program is only non-zero with respect to degrees of freedom directly involved in the collision and also respects the geometry of the contact configuration. For example, if a contact surface exists, the subgradient is orthogonal to the contact surface.

2.2.2 A Motivating Example

The aforementioned *useful additional information* is the key motivation for this work. By way of illustration, a widely accepted treatment of contact in the equations of motion may be considered: the introduction of a contact potential into the action functional. This potential takes the form of the indicator function, $I_{\mathcal{A}}$ of a set $\mathcal{A} \subset \mathcal{Q}$ containing all admissible (non-interpenetrating) configurations $\boldsymbol{q}$. Here, $\mathcal{Q}$ is a configuration manifold and $T_{\boldsymbol{q}}\mathcal{Q}$ is the tangent manifold to $\mathcal{Q}$ at $\boldsymbol{q}$, i.e. the state

varibles consist of configurations $\boldsymbol{q} \in \mathcal{Q}$ and velocities $\dot{\boldsymbol{q}} \in T_{\boldsymbol{q}}\mathcal{Q}$ (see Cirak and West [15], Clarke [16], Fetecau et al. [25], Gonzalez et al. [32], Kane et al. [38], Leyendecker et al. [50], Pandolfi et al. [68] and additional details in Chapter 3). For simplicity in the present example, both $\mathcal{Q}$ and $T_{\boldsymbol{q}}\mathcal{Q}$ will be associated with $\mathbb{R}^n$. Admissible (non-contact) configurations for $\boldsymbol{q}$ occupy the subset $\mathcal{A} \subset \mathcal{Q}$.

In the absence of other potentials and external forces, the action functional reads

$$\mathcal{I}(\boldsymbol{q}) = \int_0^T \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) dt, \tag{2.1}$$

for the Lagrangian

$$\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^T \boldsymbol{M} \dot{\boldsymbol{q}} - I_{\mathcal{A}}(\boldsymbol{q}), \tag{2.2}$$

where M is an appropriate mass matrix and

$$I_{\mathcal{A}}(\boldsymbol{q}) = \begin{cases} 0 & \text{if } \boldsymbol{q} \in \mathcal{A} \\ \infty & \text{otherwise} \end{cases}. \tag{2.3}$$

The equations of motion can be recovered by requiring stationarity of $\mathcal{I}$

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \partial I_{\mathcal{A}}(\boldsymbol{q}) \ni \boldsymbol{0}. \tag{2.4}$$

In (2.4), $\partial I_{\mathcal{A}}(\boldsymbol{q})$ denotes the *generalized differential* of the indicator function (c.f. [16, 38]). It is readily shown that the generalized differential of the indicator function of a set is given by the normal cone, $N_{\mathcal{A}}$, of the set

$$\partial I_{\mathcal{A}}(\boldsymbol{x}) = N_{\mathcal{A}}(\boldsymbol{x}). \tag{2.5}$$

It follows from (2.4) that the contact forces $\boldsymbol{f}_{con}$ are related to the normal cone by:

$$\boldsymbol{f}_{con} \in -N_{\mathcal{A}}(\boldsymbol{q}). \tag{2.6}$$

The normal cone is defined precisely in Section 2.3 of this paper. For this introductory example it is sufficient to understand that (2.6) is a statement that the contact forces must be orthogonal to

a contact surface in an admissible configuration. Alternatively, the contact time may be considered as an additional variable, leading to the action functional [15, 25]

$$\mathcal{I}(\boldsymbol{q}, t_c) = \int_0^{t_c} \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) dt + \int_{t_c}^{T} \mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}) dt. \tag{2.7}$$

Where $\mathcal{L}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ is the same as the expression in (3.34). In this case, the equations of motion at the time of contact read as jump conditions on the change of momentum $\boldsymbol{p} = \boldsymbol{M}\dot{\boldsymbol{q}}$ and kinetic energy during the collision

$$\left[\!\left[\boldsymbol{p}^T \boldsymbol{M}^{-1} \boldsymbol{p}\right]\!\right]_{t_{c^-}}^{t_{c^+}} = 0 \tag{2.8a}$$

$$[\boldsymbol{p}]_{t_{c^-}}^{t_{c^+}} \in N_{\mathcal{A}}(\boldsymbol{q}(t_c)). \tag{2.8b}$$

Equations (2.8a) and (2.8b) describe the conservation of energy and momentum during the collision, respectively. In practice, the restriction that the forces from (2.6) and the change in momentum in (2.8a) be in the normal cone of the admissible set are accomplished by constraining the configuration variables to be in $\mathcal{A} \subset \mathcal{Q}$ via an interpenetration function $g(\boldsymbol{q})$ that is negative if two bodies are not overlapping and positive if they are, that is $\mathcal{A} = \{\boldsymbol{q} \in \mathcal{Q} | g(\boldsymbol{q}) \leq 0\}$. For example, consider a point mass in two dimensions (i.e., $\boldsymbol{q} = (q_1, q_2)$) falling onto a flat surface coincident with the e_1-axis (see Figure 2.1). In this case, the admissible set of configurations for the mass are described by $\mathcal{A} = \left\{\boldsymbol{q} \in \mathbb{R}^2 | \, \langle -\hat{\boldsymbol{n}}, \boldsymbol{q} \rangle \leq 0\right\}$ with $\boldsymbol{q} = (q_1, q_2)$ and $\hat{\boldsymbol{n}} = (0, 1)^T$, and the normal cone of the admissible set has the unique values $N_{\mathcal{A}}(\boldsymbol{q}) = -\hat{\boldsymbol{n}}$ if $q_2 = 0$, $N_{\mathcal{A}}(\boldsymbol{q}) = \boldsymbol{0}$ if $q_2 > 0$ and $N_{\mathcal{A}}(\boldsymbol{q}) = \emptyset$ otherwise.

In this case, (2.8a) can be expressed as [15]

$$[\boldsymbol{p}]_{t_{c^-}}^{t_{c^+}} = \lambda \nabla g, \tag{2.9}$$

which for our example is equal to

$$[\boldsymbol{p}]_{t_{c^-}}^{t_{c^+}} = -\lambda \hat{\boldsymbol{n}} \tag{2.10}$$

if $q_2 = 0$. Here, $\lambda \in \mathbb{R}$ is a scalar parameter (see [15] for details). The simplest re-expression of (2.6) is accomplished through a smooth approximation of the indicator function

$$I_{\mathcal{A}}(\boldsymbol{q}) \approx V_{\mathcal{A}}(\boldsymbol{q}) = \begin{cases} 0 & \text{if } g(\boldsymbol{q}) < 0 \\ \frac{C}{2} g(\boldsymbol{q})^2 & \text{otherwise} \end{cases}, \tag{2.11}$$

where $C \in \mathbb{R}$ is a constant. This leads to the formulation

$$\boldsymbol{f}_{con} = -\nabla V_{\mathcal{A}}(\boldsymbol{q}), \tag{2.12}$$

which for the point-mass example is

$$\boldsymbol{f}_{con} = \begin{cases} \boldsymbol{0} & \text{if } g(\boldsymbol{q}) < 0 \\ -C \left\langle -\hat{\boldsymbol{n}}, \boldsymbol{q} \right\rangle \hat{\boldsymbol{n}} & \text{otherwise} \end{cases}. \tag{2.13}$$

This simple example lends itself to a straightforward geometric interpretation of $N_{\mathcal{A}} = \partial I_{\mathcal{A}}$, in that the contact forces must be normal to the contact surface (see (2.12)) and that the change of momentum must also be normal to the surface (see (2.9)). Thus, to accurately conserve momentum and approximate the continuous equations of motion, a constraint function $g(\boldsymbol{q})$ should have the property that $\nabla g(\boldsymbol{q}) \approx N_{\mathcal{A}}(\boldsymbol{q})$, as in the point-mass example.

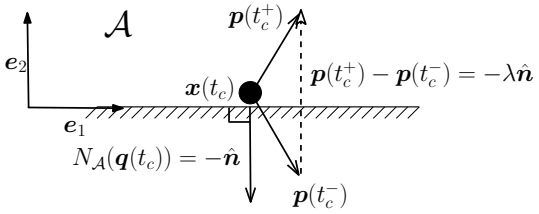


Figure 2.1: A point mass striking a flat, frictionless surface in the absence of external forces and potentials. In this example, $\lambda = 2\left\langle -\hat{\boldsymbol{n}}, \boldsymbol{p}(t_c^-)\right\rangle$.

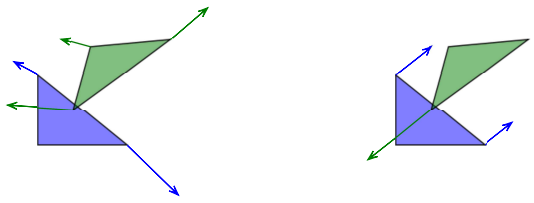
Some time integration schemes based on Lagrangian mechanics use a contact potential in the action functional and require a continuous interpenetration function that is at least sub-differentiable [15, 25, 32, 41, 50]. Such a potential is easy to construct for models of geometrically simple bodies and admissible configurations, but a good choice for this potential is much less obvious for complex geometries.

The preferred function for use in these applications has been a test for overlapping oriented simplices (OOS); i.e. tetrahedra in three dimensions, and triangles in two dimensions (c.f. [15, 38]). The OOS test can accurately provide a function $g(\boldsymbol{x})$, which indicates whether $I_{\mathcal{A}}(\boldsymbol{q})$ is zero, but it is obvious from Figure 2.2 that the gradient of the OOS test does not approximate $N_{\mathcal{A}}(\boldsymbol{q})$. This is due to

the non-global nature of the OOS test, which does not consider the body as a whole, but rather considers overlapping *segments* that compose triangles in two dimensions or overlapping *triangles* that compose tetrahedra in three dimensions. This has the further effect that the application of forces according to (2.12) may be appropriate to correct the overlap of segments in two dimensions or triangles in three dimensions, but it may not effectively correct the overlap between the bodies composed of these simplices. Thus, the OOS test is inadequate in particular for systems that do not have the capacity to absorb these spurious forces through deformations or motions; for example, in crowded or tightly-packed systems, stiff systems, and rigid body dynamics an accurate gradient $\nabla g(\boldsymbol{q}) \approx N_{\mathcal{A}}(\boldsymbol{q})$ is essential.

The OOS test has another key shortcoming. Before it can be evaluated, the algorithm must first determine whether the two segments or triangles of interest are indeed overlapping. Other versions of the test call for determining the type of contact first (face-edge, face-corner, face-face). These different 'switches' that precede the actual evaluation of the OOS test are particularly cumbersome when the time-integration algorithm is implicit or calls for optimization to resolve the contact configuration as they amount to changing the potential energy function or contact constraints as the algorithm is trying to converge.

In contrast, the supporting separating hyperplane (SSH) algorithm which is outlined in Algorithm 1 in Section 2.7 compares favorably to the OOS approach because 1) it is global in that it considers whole bodies and it does not require any 'switches' or additional calculations to classify types of contact that have taken place, and 2) it always supplies a gradient direction in the normal cone of the contact configuration, i. e., the closest admissible configuration to the present inadmissible configuration. Furthermore, the force system described by $-\nabla V_{\mathcal{A}}(\boldsymbol{q})$ in a contact configuration is local to the features on each body involved in the collision. Finally, the dual solution to the SSH linear program can be used to determine closest-feature information and to determine an excellent approximation to the exact point of contact.



(a) OOS function gradient (b) SSH function gradient

Figure 2.2: Gradient with respect to vertex locations of the OSS and SSH functions in two dimensions for a face-corner collision.

2.2.3 Organization

The development of the SSH algorithm, including the derivation of its subderivative is illustrated in the workflow diagram in Figure 2.3. The algorithm is based on concepts and theorems from convex and affine geometry, and can be reduced to a quadratically constrained linear program (QCLP) for polyhedral bodies with isolated extreme points. However, the structure of this QCLP is such that an equivalent linear program (LP) can always be formulated. Due to the nature of linear programming, an explicit expression of the subderivative of the SSH LP can be derived, which can then be used in various applications.

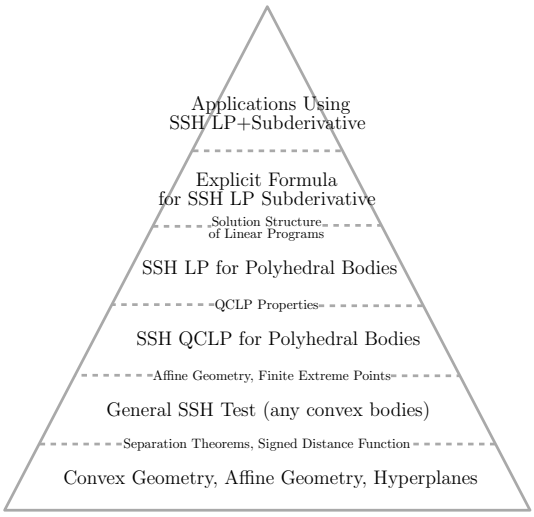


Figure 2.3: Work flow pyramid for the development of the SSH algorithm and the explicit derivation of its subderivative. Items in between levels are features of the lower level which allow progression from the lower level to the upper level.

The organization of this chapter is as follows. In Section 2.3, the basics of hyperplanes and affine geometry are reviewed, including key separation theorems for convex sets, which are used in Section 2.4 to develop the SSH test for interpenetration and prove that the algorithm always accurately indicates whether or not two convex sets are separable. In Section 2.4, it is shown that for polyhedral sets, the algorithm can always be formulated and solved as a feasible linear program. Section 2.5 reviews the solution to a linear program based on the primal simplex method that is used in Section 2.6, where nonsmooth analysis is used to expand on the results of Freund [27] and Freund [26] and derive an explicit analytical expression for the subderivative of a linear program in extended standard form. The implementation of our method is discussed in Section 2.7. Finally in Section 2.8, the effectiveness of our function in determining the e features involved in a collision and an exact collision point is illustrated.

2.3 Hyperplanes and Affine Geometry

This section provides a review of key definitions and theorems related to hyperplanes and affine sets in $\mathbb{R}^n$, which we have adapted from Rockafellar [75] and Hiriart-Urruty and Lemarechal [34], and we refer the reader to these resources for precise definitions of bounded, closed, and compact sets, and the convex hull of a set. For this and the following sections, unless otherwise stated, we denote the inner product between two vectors in $\mathbb{R}^n$ as $\langle \cdot, \cdot \rangle$, the Euclidean norm of a vector in $\mathbb{R}^n$ as $\|\cdot\|$, and the cardinality of a set as $|\cdot|$.

2.3.1 Affine Sets

A subset $M \subset \mathbb{R}^n$ is called an *affine set* if for every $\boldsymbol{x} \in M$, $y \in M$, and $\lambda \in \mathbb{R}$, $(1-\lambda)\boldsymbol{x} + \lambda \boldsymbol{y} \in M$.

2.3.2 Convex Sets

A subset $C \subset \mathbb{R}^n$ is *convex* if for all distinct points $\boldsymbol{x} \in C$, $\boldsymbol{y} \in C$, and all $0 \leq \lambda \leq 1$, $(1-\lambda)\boldsymbol{x}+\lambda \boldsymbol{y} \in C$. Clearly, all affine sets are also convex sets.

2.3.3 Hyperplanes

Following [34], we use the notation $H_{\boldsymbol{\alpha},a}$ for the hyperplane in $\mathbb{R}^n$, with $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $a \in \mathbb{R}$, as the set of points such that

$$H_{\boldsymbol{\alpha},a} := \{\boldsymbol{x} \in \mathbb{R}^n | \, \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle - a = 0\}, \tag{2.14}$$

which is an affine (and convex) set. It is recognized that $\boldsymbol{\alpha}$ is the normal vector to the plane, and that $H_{\boldsymbol{\alpha},a}$ has two distinct sides. An equivalent (point-normal) representation, with $\boldsymbol{\alpha} \in \mathbb{R}^n$ and $\boldsymbol{a} \in \mathbb{R}^n$, is

$$H_{\boldsymbol{\alpha},\boldsymbol{a}} := \{\boldsymbol{x} \in \mathbb{R}^n | \, \langle \boldsymbol{\alpha}, \boldsymbol{x} - \boldsymbol{a} \rangle = 0\}, \tag{2.15}$$

where we can identify $a = \langle \boldsymbol{\alpha}, \boldsymbol{a} \rangle$. For any choice of $\boldsymbol{\alpha}$ and $\boldsymbol{a}$, the hyperplane associated with $\mu\boldsymbol{\alpha}$ and $\boldsymbol{c} \in H_{\boldsymbol{\alpha},\boldsymbol{a}}$ is equivalent to $H_{\boldsymbol{\alpha},\boldsymbol{a}}$ for all $\mu > 0 \in \mathbb{R}$. Therefore, without loss of generality, one can

restrict $\boldsymbol{\alpha} \in S^{n-1}$. Thus, for the signed distance between a point $\boldsymbol{y} \in \mathbb{R}^n$ and a hyperplane, we use the notation

$$H_{\boldsymbol{\alpha},a}(\boldsymbol{y}) := \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle - a. \tag{2.16}$$

The definition of parallel hyperplanes can be further restricted to denote two hyperplanes $H_{\boldsymbol{\alpha},\boldsymbol{a}}$ and $H_{\mu\boldsymbol{\alpha},\boldsymbol{b}}$ for which $\mu = 1$. Thus, the signed distance from $H_{\boldsymbol{\alpha},a} \equiv H_{\boldsymbol{\alpha},\boldsymbol{a}}$ to $H_{\boldsymbol{\alpha},b} \equiv H_{\boldsymbol{\alpha},\boldsymbol{b}}$ is given by

$$d(H_{\boldsymbol{\alpha},a}, H_{\boldsymbol{\alpha},b}) = d(H_{\boldsymbol{\alpha},\boldsymbol{a}}, H_{\boldsymbol{\alpha},\boldsymbol{b}}) = b - a, \tag{2.17}$$

where the equivlance of the hyperplanes is due to $a = \langle \boldsymbol{\alpha}, \boldsymbol{a} \rangle$ and $b = \langle \boldsymbol{\alpha}, \boldsymbol{b} \rangle$, respectively.

2.3.4 Construction of Convex Sets

2.3.4.1 Intersection of Half-Spaces

Let the(closed) half-spaces associacted with a hyperplane $H_{\boldsymbol{\alpha},a}$ be defined as

$$H^+_{\boldsymbol{\alpha},a} := \{\boldsymbol{x} \in \mathbb{R}^n | H_{\boldsymbol{\alpha},a}(\boldsymbol{x}) \geq 0\} \tag{2.18a}$$

$$H^-_{\boldsymbol{\alpha},a} := \{\boldsymbol{x} \in \mathbb{R}^n | H_{\boldsymbol{\alpha},a}(\boldsymbol{x}) \leq 0\}. \tag{2.18b}$$

It may readily be shown that the intersection of convex sets is also convex (c.f. Theorem 2.1 in [75]). Thus, the following set K is convex (and closed)

$$K = \cap \{\boldsymbol{x} \in \mathbb{R}^n | H_{\boldsymbol{\alpha}_i,a_i}(\boldsymbol{x}) \leq 0, i = 1 \ldots j\}. \tag{2.19}$$

If $K \neq \emptyset$ and $j \to \infty$, then some portion of the boundary of K ($\mathrm{bd}\, K$) is said to be smooth. Furthermore, K is a compact set if it is bounded. For finite j, K as defined above is non-smooth and is called a polyhedral set. In the case that a polyhedral set is bounded (and therefore compact), it may alternately be called a polytope in $\mathbb{R}^n$ or simply a polyhedral body (also in $\mathbb{R}^n$), in reference to the physical applications of the algorithm to be developed in this work.

2.3.4.2 Convex Hull of Extreme Points

An extreme point $\boldsymbol{z} \in K$ is a point such that there are no two different points $\boldsymbol{x} \in K$ and $\boldsymbol{y} \in K$ for which $\boldsymbol{z} = (1-\gamma)\boldsymbol{x} + \gamma\boldsymbol{y}, 0 \leq \gamma \leq 1$. That is, $\boldsymbol{z}$ is an extreme point of K if and only if $\boldsymbol{z} = (1-\gamma)\boldsymbol{x} + \gamma\boldsymbol{y}, 0 \leq \gamma \leq 1 \Rightarrow \boldsymbol{x} = \boldsymbol{y} = \boldsymbol{z}$. Let us denote the set of extreme points of K as $\operatorname{ext} K$. For a compact set, $\operatorname{ext} K \neq \emptyset$, and all extreme points of K are on $\operatorname{bd} K$. Thus, a set K, which is convex and compact in $\mathbb{R}^n$, can alternately be described as the convex hull of its extreme points, $K = \operatorname{co}(\operatorname{ext} K)$. By this construction, for a finite number of extreme points, K is a polytope. This allows the description of all points in K as a convex combination of its extreme points. That is, for all $\boldsymbol{x} \in K$, $\boldsymbol{y} \in \operatorname{ext} K$, and $\lambda_i > 0$,

$$\begin{aligned} \boldsymbol{x} = &\sum_{i}^{|\operatorname{ext} K|} \lambda_i \boldsymbol{y}_i, \text{ where:} \\ &\sum_{i}^{|\operatorname{ext} K|} \lambda_i = 1. \end{aligned} \tag{2.20}$$

For a polyhedral set, the set of vertices is equal to the set of extreme points.

2.3.5 Supporting Hyperplanes

A hyperplane $H_{\boldsymbol{\alpha},\boldsymbol{a}}$ is said to support the set K when K is entirely contained in either $H^+_{\boldsymbol{\alpha},a}$ or $H^-_{\boldsymbol{\alpha},a}$, and $\operatorname{bd} K \cap H_{\boldsymbol{\alpha},a} \neq \emptyset$. It is said to support K at $\boldsymbol{x} \in K$ when, in addition, $\boldsymbol{x} \in H_{\boldsymbol{\alpha},\boldsymbol{a}}$.

2.3.6 Normal Cone to a Convex Set

For the following analysis, it suffices to present a normal cone definition that is strictly relevant to convex sets. However, it is important to note that the idea can be extended to non-convex sets and is important in non-smooth analysis (see Clarke [16] for additional information). The normal cone $N_K(\boldsymbol{x})$ to a convex set K at the point $\boldsymbol{x} \in K$ may be defined as the set of directions $\boldsymbol{\nu} \in \mathbb{R}^n$ for which K is in the negative halfspace of $H_{\boldsymbol{\nu},\boldsymbol{x}}$, i.e.

$$\begin{aligned} N_K(\boldsymbol{x}) &= \{\boldsymbol{\nu} \in \mathbb{R}^n | H_{\boldsymbol{\nu},\boldsymbol{x}}(\boldsymbol{x}^*) \leq 0 \ \forall \boldsymbol{x}^* \in K\} \\ N_K(\boldsymbol{x}) &= \left\{\boldsymbol{\nu} \in \mathbb{R}^n | K \subset H^-_{\boldsymbol{\nu},\boldsymbol{x}}\right\}. \end{aligned} \tag{2.21}$$

Thus, it is obvious that $N_K(\boldsymbol{x})$ will contain more than one direction $\boldsymbol{\nu}$ when K is a polyhedral set of dimension n in $\mathbb{R}^n$ if $\boldsymbol{x}$ is in a feature of K of dimension less than $n-1$. Figure 2.4 shows two examples of normal cones.

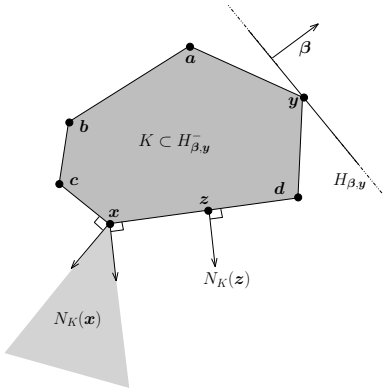


Figure 2.4: Schematic of concepts for hyperplanes and convex sets for a compact polyhedral set, K. Points $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{x}, \boldsymbol{d}, \boldsymbol{y}$ form the set of extreme points of K, and $\boldsymbol{z}$ is not an extreme point. The hyperplane $H_{\boldsymbol{\beta},\boldsymbol{y}}$ supports the set K at $\boldsymbol{y}$ such that $K \subset H^-_{\boldsymbol{\beta},\boldsymbol{y}}$. The normal cone of K is shown at two points: $\boldsymbol{x} \in \operatorname{ext} K$ where $N_K(\boldsymbol{x})$ is not unique, and $\boldsymbol{z} \notin \operatorname{ext} K$ where $N_K(\boldsymbol{x})$ is unique.

As it will be useful later, it is noted here that for convex sets there is a one-to-one correspondence between finding a hyperplane supporting a set at a given point, and finding a direction in the normal cone at the point.

2.3.7 Separation of Convex Sets

The *strict* separation of sets is defined as follows: for two non-empty closed convex sets, K_1 and K_2 for which $K_1 \cap K_2 = \emptyset$ and K_2 is bounded, there exists $\boldsymbol{\alpha}$ such that:

$$\sup_{\boldsymbol{y} \in K_2} \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle < \min_{\boldsymbol{x} \in K_1} \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle . \tag{2.22}$$

Furthermore, if K_1 is also bounded, then there exists $\boldsymbol{\alpha}$ such that

$$\max_{\boldsymbol{y} \in K_2} \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle < \min_{\boldsymbol{x} \in K_1} \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle . \tag{2.23}$$

Two compact convex sets are *properly* separated if:

$$\max_{\boldsymbol{y}\in K_2} \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle \leq \min_{\boldsymbol{x}\in K_1} \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle \,, \text{and} \tag{2.24a}$$

$$\min_{\boldsymbol{y}\in K_2} \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle < \max_{\boldsymbol{x}\in K_1} \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle \,. \tag{2.24b}$$

In other words, a vector $\boldsymbol{\alpha}$ is a separating vector if it can be associated with a hyperplane $H_{\boldsymbol{\alpha},a}$ that properly classifies all of the points K_1 and K_2 such that $K_1 \subset H^+_{\boldsymbol{\alpha},a}$ and $K_2 \subset H^-_{\boldsymbol{\alpha},a}$. The converse is also true. For two compact convex sets, if $\boldsymbol{\alpha}$ cannot be found such that $\max_{\boldsymbol{y}\in K_2} \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle \leq \min_{\boldsymbol{x}\in K_1} \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle$, then $K_1 \cap K_2 \neq \emptyset$. Figure 2.5 illustrates the definition of proper ($\boldsymbol{\alpha}$) and strict ($\boldsymbol{\beta}$) separating vectors, along with examples of hyperplanes associated with these vectors.

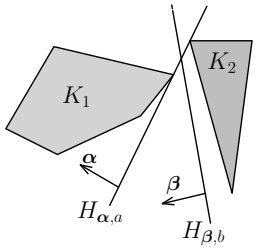


Figure 2.5: An example of strictly separable sets. The vector $\boldsymbol{\beta}$ and associated hyperplane $H_{\boldsymbol{\beta},b}$ strictly separate K_1 and K_2, whereas the vector $\boldsymbol{\alpha}$ and associated hyperplane $H_{\boldsymbol{\alpha},a}$ properly separate K_1 and K_2. Note that the hyperplane drawn labeled $H_{\boldsymbol{\beta},b}$ is not a unique choice for the vector $\boldsymbol{\beta}$.

2.4 Supporting Separating Hyperplane Algorithm

In this section, the Supporting Separating Hyperplane (SSH) algorithm as a test for interpenetration is developed. To begin, the algorithm is developed for general compact convex sets as a constrained optimization problem with a linear objective function. It is then shown that, for compact polyhedral sets, the algorithm can always be reformulated so that all constraints are linear, i.e., as a linear programming problem.

To begin, let $\{H_{\boldsymbol{\alpha},a}\}^+_{K_1}$ be defined as the set of supporting hyperplanes of K_1 at $\boldsymbol{x} \in \operatorname{bd} K_1$ such that $K_1 \subset H^+_{\boldsymbol{\alpha},a}$ and $\{H_{\boldsymbol{\beta},b}\}^-_{K_2}$ be defined as the set of supporting hyperplanes of K_2 at $\boldsymbol{y} \in \operatorname{bd} K_2$ such that $K_2 \subset H^-_{\boldsymbol{\beta},b}$. Again, all normal vectors of $\{H_{\boldsymbol{\alpha},a}\}^+_{K_1}$ and $\{H_{\boldsymbol{\beta},b}\}^-_{K_2}$ are unit vectors (i.e., $\boldsymbol{\alpha}, \boldsymbol{\beta} \in S^{n-1}$). Define the function $\tilde{h}(K_1, K_2)$ as follows, where the function $d(H_{\boldsymbol{\alpha},a}, H_{\boldsymbol{\alpha},b})$ was defined in (2.17) and is the distance between parallel hyperplanes:

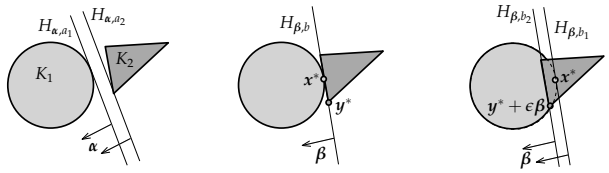
$$\tilde{h}(K_1, K_2) = \max_{\substack{H_{\boldsymbol{\gamma},\bar{a}} \in \{H_{\boldsymbol{\alpha},a}\}^+_{K_1} \\ H_{\boldsymbol{\gamma},\bar{b}} \in \{H_{\boldsymbol{\beta},b}\}^-_{K_2}}} d(H_{\boldsymbol{\gamma},\bar{a}}, H_{\boldsymbol{\gamma},\bar{b}}) \tag{2.25}$$

The solution to (2.25) is the maximum signed distance between parallel supporting hyperplanes of each set, with the restriction that each hyperplane properly classify its associated set.

Remark 2.1. *From the point-normal description of a hyperplane, for two non-empty compact convex sets,* $\tilde{h}(K_1, K_2) = \tilde{h}(\operatorname{ext} K_1, \operatorname{ext} K_2)$. *Since* $d(H_{\boldsymbol{\alpha},\boldsymbol{a}_2}, H_{\boldsymbol{\alpha},\boldsymbol{a}_1}) = d(H_{\boldsymbol{\alpha},\boldsymbol{a}_2}, H_{\boldsymbol{\alpha},\boldsymbol{b}_1})\ \forall \boldsymbol{b}_1 - \boldsymbol{a}_1 \in H_{\boldsymbol{\alpha},\boldsymbol{a}_1}$, *i.e.,* $\boldsymbol{b}_1 - \boldsymbol{a}_1 \in H_{\boldsymbol{\alpha},\boldsymbol{a}_1} \Rightarrow H_{\boldsymbol{\alpha},\boldsymbol{a}_1} \cap H_{\boldsymbol{\alpha},\boldsymbol{b}_1} = H_{\boldsymbol{\alpha},\boldsymbol{a}_1} = H_{\boldsymbol{\alpha},\boldsymbol{b}_1}$. *Therefore, if* $H_{\boldsymbol{\alpha},\boldsymbol{a}_1}$ *supports* K_1 *at* $\boldsymbol{a}_1 \notin \operatorname{ext} K_1$, *there is an equivalent hyperplane that supports* K_1 *at* $\boldsymbol{b}_1 \in \operatorname{ext} K_1$, *and*

$$\tilde{h}(\operatorname{ext} K_1, \operatorname{ext} K_2) = \tilde{h}(K_1, K_2). \tag{2.26}$$

Remark 2.2. *The problem (2.25) is always feasible. For two compact, convex sets, one can always find two parallel supporting hyperplanes such that* $H_{\boldsymbol{\alpha},a_1} \in \{H_{\boldsymbol{\alpha},a}\}^+_{K_1}$ *and* $H_{\boldsymbol{\alpha},a_2} \in \{H_{\boldsymbol{\beta},b}\}^-_{K_2}$. *This remark is obvious if* $K_1 \cap K_2 = \emptyset$, *and at least one hyperplane exists that strictly separates* K_1 *and* K_2. *To show that this is true if* $K_1 \cap K_2 \neq \emptyset$, *consider a plane* $H_{\boldsymbol{\beta},b}$ *that properly separates two sets for which* $\max_{\boldsymbol{y} \in K_2} \langle \boldsymbol{\beta}, \boldsymbol{y} \rangle = \min_{\boldsymbol{x} \in K_1} \langle \boldsymbol{\beta}, \boldsymbol{x} \rangle$, *and* K_1 *and* K_2 *uniquely properly separable. Because the sets are compact and convex, such a plane can always be found. Further, define two equivalent planes for which* $H_{\boldsymbol{\beta},b_2}(\boldsymbol{y}^*) = 0$ *for some* $\boldsymbol{y}^* \in \operatorname{bd} K_2$ *and* $H_{\boldsymbol{\beta},b_1}(\boldsymbol{x}^*) = 0$ *for some* $\boldsymbol{x}^* \in \operatorname{bd} K_1$. *If one now sets* $b_2 = \langle \boldsymbol{\beta}, \boldsymbol{y}^* \rangle$ *and* $b_1 = \langle \boldsymbol{\beta}, \boldsymbol{x}^* \rangle$ *and undertakes a translation of* K_2 *and, by extension,* $H_{\boldsymbol{\beta},b_2}$ *by* $\epsilon\boldsymbol{\beta}$ *so that* $b_2 = \langle \boldsymbol{\beta}, \boldsymbol{y}^* + \epsilon\boldsymbol{\beta} \rangle$, K_1 *and* K_2 *no longer are separable, but two parallel hyperplanes that are in the domain of (2.25) are retained. The cases discussed in this remark are illustrated in Figure 2.6.*



(a) Strictly separable sets (b) Properly separable sets (c) Non-separable sets

Figure 2.6: Figures 2.6a through 2.6c illustrate the cases discussed in Remark 2.2. One of the many possible sets of supporting hyperplanes in the domain of (2.25) is shown for strictly separable sets in Figure 2.6a. In Figures 2.6b and 2.6c, the circular set remains K_1, and the triangular set remains K_2, but they are not explicitly labeled to simplify the drawing. Figure 2.6b shows a hyperplane ($H_{\boldsymbol{\beta},b}$) which is clearly in the domain of (2.25), and two arbitrarily chosen points, $\boldsymbol{x}^* \in \operatorname{bd} K_1$ and $\boldsymbol{y}^* \in \operatorname{bd} K_2$, at which $H_{\boldsymbol{\beta},b}$ supports each set, as described in Remark 2.2. Finally, Figure 2.6c shows the translation of K_2 by $\epsilon\boldsymbol{\beta}$, and the corresponding hyperplanes in the domain of (2.25).

Before proceeding, a constrained optimization problem over $\boldsymbol{\alpha} \in S^{n-1}$, and $a_1, a_2 \in \mathbb{R}$ that has the

same solution to (2.25) and is useful in proving Theorem 2.1 can be defined as

$$\begin{aligned} h(K_1, K_2) = \max_{\substack{\boldsymbol{\alpha} \in S^{n-1} \\ a_1, a_2 \in \mathbb{R}}} & a_1 - a_2, \text{ Subject to:} \\ & \operatorname{ext} K_1 \subset H^+_{\boldsymbol{\alpha}, a_1} \\ & \operatorname{ext} K_2 \subset H^-_{\boldsymbol{\alpha}, a_2}. \end{aligned} \tag{2.27}$$

Theorem 2.1. *Two compact convex sets K_1 and K_2 are strictly (properly) separable if and only if $h > (\geq) 0$.*

Proof. *It is first shown that $h > 0 \Rightarrow K_1 \cap K_2 = \emptyset$, and by extension that $h = 0$ implies that K_1 and K_2 are properly separable. From the definition of a supporting hyperplane, $a_1 \leq \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle$ for $\boldsymbol{x} \in \operatorname{ext} K_1$ and likewise $a_2 \geq \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle$ for $\boldsymbol{y} \in \operatorname{ext} K_2$. It follows from the constraints in equation (2.27) that $\max a_1 = \min_{\boldsymbol{x} \in \operatorname{ext} K_1} \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle$ and that $\min a_2 = \max_{\boldsymbol{y} \in \operatorname{ext} K_2} \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle$ at the solution. Therefore, we can do a portion of the optimization explicitly and see that*

$$\begin{aligned} h(K_1, K_2) &= \max_{\substack{\boldsymbol{\alpha} \in S^{n-1} \\ \boldsymbol{x} \in \operatorname{ext} K_1 \\ \boldsymbol{y} \in \operatorname{ext} K_2}} a_1(\boldsymbol{\alpha}, \boldsymbol{x}) - a_2(\boldsymbol{\alpha}, \boldsymbol{y}) \\ &= \min_{\substack{\boldsymbol{\alpha} \in S^{n-1} \\ \boldsymbol{x} \in \operatorname{ext} K_1}} \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle - \max_{\substack{\boldsymbol{\alpha} \in S^{n-1} \\ \boldsymbol{y} \in \operatorname{ext} K_2}} \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle, \end{aligned}$$

subject to the same constraints as (2.27). From the definition of strict (proper) separation of sets, if $\boldsymbol{\alpha}$ can be found such that $h(K_1, K_2) > (\geq) 0$, then K_1 and K_2 are strictly (properly) separable, thus $h > 0 \Rightarrow K_1 \cap K_2 = \emptyset$, and that $h = 0 \Rightarrow K_1 \cap K_2 \subset H_{\boldsymbol{\alpha}, a_1} \equiv H_{\boldsymbol{\alpha}, a_2}$.

To show that $h > 0 \Leftarrow K_1 \cap K_2 = \emptyset$, note that the domain has been restricted such that $H_{\boldsymbol{\alpha}, a_1}$ properly classifies $K_1 \subset H^+_{\boldsymbol{\alpha}, a_1}$, and $H_{\boldsymbol{\alpha}, a_2}$ properly classifies $K_2 \subset H^-_{\boldsymbol{\alpha}, a_1}$. In this domain, the maximum distance $a_1 - a_2$ is positive (non-negative) only if, in addition, $K_2 \subset H^-_{\boldsymbol{\alpha}, a_1}$, and $K_1 \subset H^+_{\boldsymbol{\alpha}, a_2}$, in which case, by the definition of strict (proper) separation of sets, $\boldsymbol{\alpha}$ is a separating vector and both hyperplanes are separating hyperplanes. Therefore $h > 0 \Leftarrow K_1 \cap K_2 = \emptyset$, and $h = 0 \Leftarrow K_1 \cap K_2 \subset H_{\boldsymbol{\alpha}, a_1} \equiv H_{\boldsymbol{\alpha}, a_2}$. □

Corallary 2.2. *Corallary to Theorem 2.1 Two compact convex sets K_1 and K_2 are not separable if and only if $h < 0$.*

Proof. *Follows directly from Theorem 2.1.* □

Figure 2.7 is an illustration of several concepts in the proof of Theorem 2.1 and its corollary.

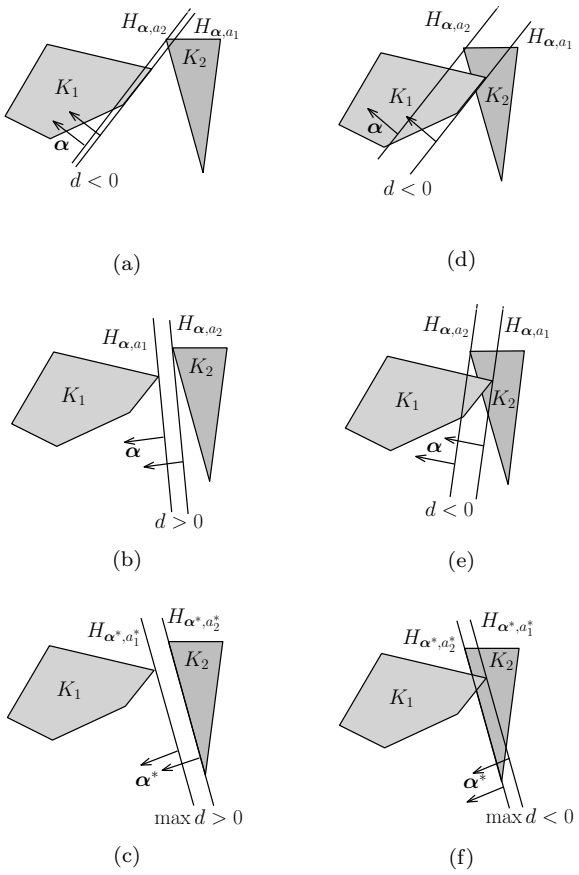


Figure 2.7: Figures 2.7a through 2.7c show the domain of *potentially separating* supporting hyperplanes for two separable polyhedral sets. The sign of d denotes whether $H_{\boldsymbol{\alpha},a_1}$ is in front of (positive) or behind (negative) $H_{\boldsymbol{\alpha},a_2}$ according to the direction of α. Figures 2.7d through 2.7f show the same domain for sets that are not separable.

2.4.1 Formulation as a Quadratically Constrained Linear Program

Equation 2.27 can further be refined by transforming the restriction on the domain to constraints on the solution vector according to the definition of a supporting hyperplane as follows:

$$\begin{aligned} h(K_1, K_2) = \max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^n \\ a_1,a_2\in\mathbb{R}}} \; & a_1 - a_2, \text{ Subject to:} \\ & \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle - a_1 \geq 0, \; \boldsymbol{x} \in \operatorname{ext} K_1 \\ & \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle - a_2 \leq 0, \; \boldsymbol{y} \in \operatorname{ext} K_2 \\ & \langle \boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle = 1. \end{aligned} \tag{2.28}$$

Note that if K_1 and K_2 are both polytopes, then there is exactly one constraint in (2.28) that is not

linear. This constraint corresponds to the restriction of $\boldsymbol{\alpha}$ to be in S^{n-1}. In the following section it is shown that this constraint may be relaxed (to a restriction that $H_{\hat{\boldsymbol{\beta}},1}(\boldsymbol{\alpha}) = 0$) without changing the essential result that $h > 0 \Leftrightarrow K_1 \cap K_2 = \emptyset$ and $h \leq 0 \Leftrightarrow K_1 \cap K_2 \neq \emptyset$.

2.4.2 Interpenetration Detection as a Linearly Constrained Linear Program

The goal of this section is to formulate a linear program (LP) which, like (2.28) indicates whether or not two sets are intersecting when the sets K_1 and K_2 are both compact polyhedral sets. To begin, we note that the structure of (2.28) is linear in the objective function, and in all of the constraints with the exception of the final constraint, that $\langle \boldsymbol{\alpha}, \boldsymbol{\alpha} \rangle = 1$. For the purposes of this analysis, consideration will be restricter to sets for which $\dim(K_1) = \dim(K_2) = n$, i.e. sets of the same dimension of the space in which they are embedded, however the results may be extended to some cases in which $\dim(K_1) < n$ and/or $\dim(K_2) < n$. The following remarks will be useful in showing that the solution to a linear program of the form

$$
\begin{aligned}
g(K_1, K_2) = \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^n \\ a_1, a_2 \in \mathbb{R}}} \; & a_1 - a_2, \text{ Subject to:} \\
& \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle - a_1 \geq 0, \; \boldsymbol{x} \in \operatorname{ext} K_1 \\
& \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle - a_2 \leq 0, \; \boldsymbol{y} \in \operatorname{ext} K_2 \\
& \langle \boldsymbol{\beta}, \boldsymbol{\alpha} \rangle = 1,
\end{aligned}
\tag{2.29}
$$

for certain choices of $\boldsymbol{\beta}$ is equivalent to (2.28), in that a solution vector, $\overline{\boldsymbol{a}} = (\boldsymbol{\alpha} \; a_1 \; a_2)$ to the quadratically constrained program (the QCLP, or simply QP) is equal to a solution to the linearly constrained program (the LP) normalized by $\|\boldsymbol{\alpha}\|$.

Remark 2.3. *For two compact convex sets, K_1 and K_2, if $K_1 \cap K_2 \neq \emptyset$, then $d(H_{\boldsymbol{\alpha},a_1}, H_{\boldsymbol{\alpha},a_2}) \leq 0 \, \forall \, H_{\boldsymbol{\alpha},a_1} \in \{H_{\boldsymbol{\alpha},a}\}^+_{K_1}$, $H_{\boldsymbol{\alpha},a_2} \in \{H_{\boldsymbol{\alpha},a}\}^-_{K_2}$. This follows from Theorem 2.1 due to $0 \geq \max d(H_{\boldsymbol{\alpha},a_2}, H_{\boldsymbol{\alpha},a_2}) \geq d(H_{\boldsymbol{\alpha},a_2}, H_{\boldsymbol{\alpha},a_2})$.*

Remark 2.4. *If two compact, convex sets are separable with $\dim(K_1) = \dim(K_2) = n$, then there is a compact set of directions $\boldsymbol{\alpha} \in S^{n-1}$ which render the QP (2.28) non-negative. This follows from the observation that there is a compact set of points in $\operatorname{bd} K_1$ and $\operatorname{bd} K_2$ at which a separating hyperplane can support each set. Furthermore, with the restriction that $\dim(K_1) = \dim(K_2) = n$, this set will be contained in less than a hemisphere of S^{n-1}. This set of separating directions will be denoted as $\mathcal{S}(K_1, K_2) = \{\boldsymbol{\alpha} \in S^{n-1} \mid$*

$K_1 \subset H^+_{\boldsymbol{\alpha},\boldsymbol{x}}$, $K_2 \subset H^-_{\boldsymbol{\alpha},\boldsymbol{y}}$, $\boldsymbol{x} \in \operatorname{bd} K_1$, $\boldsymbol{y} \in \operatorname{bd} K_2\} \subset S^{n-1}$. *An example of* $\mathcal{S}(K_1, K_2)$ *for two sets in* $\mathbb{R}^2$ *is shown in Figure 2.8.*

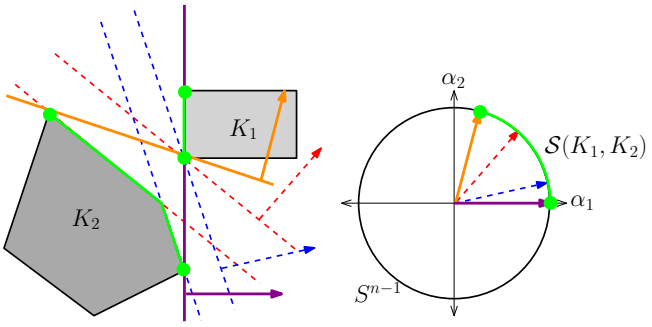


Figure 2.8: The left schematic highlights the set of points in $\operatorname{bd} K_1$ and $\operatorname{bd} K_2$ at which each set can be supported, along with several pairs of supporting hyperplanes satisfying the constraints of (2.28). In the right schematic, the set $\mathcal{S}(K_1, K_2)$ of separating directions for the two bodies sketched in the left schematic is shown.

Remark 2.5. *The QP (2.28) always has a finite solution so long as all* $\boldsymbol{x}_i$, *and* $\boldsymbol{y}_i$ *are finite. This follows from 1) the observation that the objective function only has a non-zero gradient in the* $a_1 - a_2$ *subspace, and 2) the structure of the first two sets of contraints establishes an upper bound for* a_1, *and a lower bound for* a_2 *for any* $\boldsymbol{\alpha}$. *Thus, the objective function is always bounded in its increasing direction for any* $\boldsymbol{\alpha} \in S^{n-1}$, *as can be seen in Figure 2.9.*

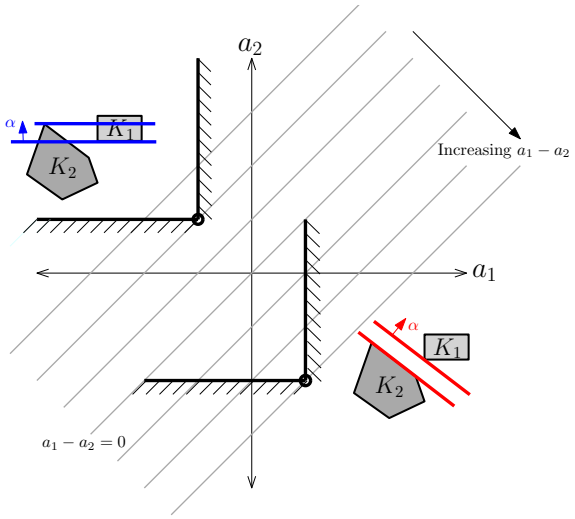


Figure 2.9: The level sets of the objective function, $a_1 - a_2$, are shown. Schematics of the upper bound for a_1 and the lower bound for a_2 are shown for selected directions $\boldsymbol{\alpha}$ which are feasible solutions to (2.28), sketched in the correponding diagrams superimposed on the plot.

Remark 2.6. *The LP (2.29) is feasible for any choice of* $\boldsymbol{\beta}$. *To show this, note that the following problem is equivalent to (2.28)*

$$h(K_1, K_2) = \max_{\boldsymbol{\beta} \in S^{n-1}} \max_{\substack{\boldsymbol{\alpha} \in S^{n-1} \\ a_1, a_2 \in \mathbb{R}}} a_1 - a_2, \text{ Subject to:}$$

$$\langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle - a_1 \geq 0, \ \boldsymbol{x} \in \operatorname{ext} K_1$$

$$\langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle - a_2 \leq 0, \ \boldsymbol{y} \in \operatorname{ext} K_2$$

$$\langle \boldsymbol{\alpha}, \boldsymbol{\beta} \rangle - 1 = 0.$$

The inner optimization problem then amounts to an optimization over (a_1, a_2) for a fixed vector $\boldsymbol{\alpha} = \bar{\boldsymbol{\beta}}$, which is always a bounded, feasible problem according to the preceeding remark. Thus, removing the constraint that $\boldsymbol{\alpha} \in S^{n-1}$ will result in a problem with at least one feasible solution for any $\bar{\boldsymbol{\beta}}$.

With these preliminaries established, call $a_1 - a_2 = \gamma$ a feasible solution to the QP (2.28), and $a_1^* - a_2^* = \gamma^*$ an optimal solution to the QP. The corresponding equivalent solutions to the LP (2.29) will have the values $\gamma \|\boldsymbol{\alpha}_{LP}\|$ and $\gamma^* \|\boldsymbol{\alpha}_{LP}^*\|$, respectively, with $\boldsymbol{\alpha}_{LP}$ satisfying $H_{\boldsymbol{\beta},1}(\boldsymbol{\alpha}_{LP}) = 0$. Futhermore, we will use the notation $\boldsymbol{a} = (a_1, a_2)$ to denote the solution to the QP (2.28) is the $a_1 - a_2$ subspace, and $\boldsymbol{a}_{LP}$ the corresponding solution to the LP (2.29), with the relationship $\boldsymbol{a} = \frac{\boldsymbol{a}_{LP}}{\|\boldsymbol{\alpha}_{LP}\|}$. Before proceeding with the key theorem of this section, recall the definition of the set of separating vectors for the sets given in Remark 2.4 as $\mathcal{S}(K_1, K_2) = \{\boldsymbol{\alpha} \in S^{n-1} |\ K_1 \subset H^+_{\boldsymbol{\alpha}, \boldsymbol{x}},\ K_2 \subset H^-_{\boldsymbol{\alpha}, \boldsymbol{y}},$ $\boldsymbol{x} \in \operatorname{bd} K_1,\ \boldsymbol{y} \in \operatorname{bd} K_2\} \subset S^{n-1}$, which will be equal to the empty set if K_1 and K_2 are not separable.

Theorem 2.3. *An optimal solution, $\overline{\boldsymbol{a}}_{LP}^*$, to (2.29) is equivalent to an optimal solution, $\overline{\boldsymbol{a}}_{QP}^*$, to (2.28) for a given choice of $\boldsymbol{\beta} \in S^{n-1}$ if the following statements hold.*

i The open hemisphere of S^{n-1} centered on $\boldsymbol{\beta}$ contains of $\mathcal{S}(K_1, K_2)$ and $\boldsymbol{\alpha}_{QP}^*$.

ii For all other feasible solutions to the linear program, $\overline{\boldsymbol{a}}_{LP}^i$, $\operatorname{sign}(\gamma^i) \neq \operatorname{sign}(\gamma^*)$, $\left\|\boldsymbol{\alpha}_{LP}^i\right\| < \left|\frac{\gamma^* \|\boldsymbol{\alpha}_{LP}^*\|}{\gamma^i} - 1\right| + 1$ for $\gamma^* > 0$, or $\left\|\boldsymbol{\alpha}_{LP}^i\right\| > \left|\frac{\gamma^* \|\boldsymbol{\alpha}_{LP}^*\|}{\gamma^i} - 1\right| + 1$ for $\gamma^* < 0$.

Proof. *Proof of Theorem 2.3*

By statement (i), the solution direction $\boldsymbol{\alpha}_{QP}^*$ has an equivalent solution direction $\boldsymbol{\alpha}_{LP} \in H_{\boldsymbol{\beta},1}$. Also by statement (i), as $\|\boldsymbol{\alpha}\| \to \infty$, γ becomes negative for directions $\boldsymbol{\alpha} \in H_{\boldsymbol{\beta},1}$, so all feasible solutions to the LP are bounded from above. It is still possible for another feasible solution direction, $\boldsymbol{\alpha}_{LP}^0 \in H_{\boldsymbol{\beta},1}$, of the linear program to satisfy $\gamma^0 \left\|\boldsymbol{\alpha}_{LP}^0\right\| > \gamma^* \|\boldsymbol{\alpha}_{LP}^*\|$, with $\gamma^0 < \gamma^*$. Trivially, if $\operatorname{sign}(\gamma^0) \neq \operatorname{sign}(\gamma^*)$, then $\boldsymbol{\alpha}_{LP}^*$ remains the optimal LP direction. If $\operatorname{sign}(\gamma^0) = \operatorname{sign}(\gamma^*)$, then we

can calculate the point, $\boldsymbol{a}_{crit}$*, at which the ray* $a_2 = \frac{a_2^0}{a_1^0} a_1$ *(with* $\boldsymbol{a}^0 = (a_1^0, a_2^0)$ *as components of the corresponding feasible solution to (2.28)) intersects the level set* $a_1 - a_2 = \gamma^* \left\| \boldsymbol{\alpha}_{LP}^* \right\|$ *as*

$$\boldsymbol{a}_{crit} = \frac{\gamma^* \left\| \boldsymbol{\alpha}_{LP}^* \right\|}{\gamma^0} \boldsymbol{a}^0$$

It then follows from $\left\| \boldsymbol{\alpha}_{LP}^0 \right\| = \frac{\left\| \boldsymbol{a}_{LP}^0 - \boldsymbol{a}^0 \right\|}{\left\| \boldsymbol{a}^0 \right\|} + 1$ *that*

$$\left\| \boldsymbol{\alpha}_{LP}^0 \right\| < \left\| \boldsymbol{\alpha}_{crit} \right\| \Rightarrow \gamma^0 \left\| \boldsymbol{\alpha}_{LP}^0 \right\| < \gamma^* \left\| \boldsymbol{\alpha}_{LP}^* \right\| \; for \gamma^* > 0,$$

$$and$$

$$\left\| \boldsymbol{\alpha}_{LP}^0 \right\| > \left\| \boldsymbol{\alpha}_{crit} \right\| \Rightarrow \gamma^0 \left\| \boldsymbol{\alpha}_{LP}^0 \right\| < \gamma^* \left\| \boldsymbol{\alpha}_{LP}^* \right\| \; for \gamma^* < 0, \qquad \square$$

with $\left\| \boldsymbol{\alpha}_{crit} \right\| = \left| \frac{\gamma^* \left\| \boldsymbol{\alpha}_{LP}^* \right\|}{\gamma^0} - 1 \right| + 1$*, which is the final condition in statement (ii).*

In practice, choosing a vector $\boldsymbol{\beta}$ a priori to meet the first requirement of Theorem 2.3 is not hard to do, in particular as the optimal QP solution decreases to zero or becomes negative and $\mathcal{S}(K_1, K_2)$ shrinks to the empty set, which renders (i) less and less restrictive. We will not present a concise result on this topic, although we note that $\mathcal{S}(K_1, K_2)$ could be explicitly calculated and a central direction in this set chosen if needed to ensure that (i) is met so that the optimal LP solution is bounded. Furthermore, if an unbounded LP is encountered–which in itself implies that the sets are separable–$\boldsymbol{\beta}$ can simply be reset based on the unbounded direction, and the new problem solved to determine a separating direction.

Perhaps more importantly, choosing $\boldsymbol{\beta}$ to meet the second requirement of the theorem is actually not essential to determine whether or not two sets are separable, or even to resolve a separating direction if they are. By Remark 2.3, if K_1 and K_2 are not separable, then *any* choice of $\boldsymbol{\beta}$ will indicate this fact, by rendering the objective function of the LP (2.29) negative. However, it is possible to select $\boldsymbol{\beta}$ which gives a 'false positive', for which the solution to the LP is negative, but the solution to the QP is positive, implying that condition (i) in Theorem 2.3 has not been met. To avoid this, $\boldsymbol{\beta}$ should be initially selected so that $\boldsymbol{\beta}$ and some subset $\bar{\mathcal{S}} \subset \mathcal{S}(K_1, K_2)$ are in the same open hemisphere.

In other words, one does not need to find an *equivalent* LP solution to the QP (2.28) in order to obtain an accurate result for the SSH test using a linear program. Rather, one simply needs to solve a bounded LP which avoids the aforementioned 'false positive'. A good a priori choice for a vector

$\boldsymbol{\beta}$ satisfying condition (i) of Theorem 2.3–or barring that, a direction such that $\boldsymbol{\beta}$ and $\bar{\mathcal{S}}$ are in the same open hemisphere–is a unit vector along a direction connecting a point $\boldsymbol{y} \in K_2$ to another point $\boldsymbol{x} \in K_1$. One such selection is presented in Section 2.7.4.

2.5 Linear Programming

For completeness, this section provides a brief review of the well-studied topic of optimality conditions and solution strategies for linear programs (Dantzig and Thapa [19] provides a thorough primary reference). While this will likely be a review for many readers, the formulation of the problem is essential to the evaluation of its SSH linear program's subderivative.

To simplify notation in this section, the number of inequality constraints in a linear program whill be denoted as m_1, the number of equality constraints as m_2, and the number of primal variables is n. All vectors are to be understood as column vectors and are denoted by bold lowercase letters, for example $\boldsymbol{a} \in \mathbb{R}^n$. The combination of two vectors $\boldsymbol{a}_1 \in \mathbb{R}^{n_1}$ and $\boldsymbol{a}_2 \in \mathbb{R}^{n_2}$ into an extended vector $\boldsymbol{a} \in \mathbb{R}^{n_1+n_2}$ is denoted, with a slight abuse of notation, as $\boldsymbol{a} = \begin{pmatrix} \boldsymbol{a}_1 & \boldsymbol{a}_2 \end{pmatrix}$. Subscript notation such as a_i denotes the i^{th} component of $\boldsymbol{a}$. Matrices such as $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ are denoted by bold uppercase letters, and the combination of matrices of compatible dimensions into entries in block matrices is denoted by means of square brackets, e.g. $\begin{bmatrix} \boldsymbol{A}_1 & \boldsymbol{A}_2 \end{bmatrix}$. Subscript notation such as A_{ij} denotes the j^{th} component of the i^{th} row of $\boldsymbol{A}$.

2.5.1 Conditions for Optimality

Linear programs of the following form will be considered

$$\begin{aligned} F(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) &= \max_{\boldsymbol{x} \in X} \boldsymbol{c}^T \boldsymbol{x} \\ \text{with: } X &= \{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{A}_1 \boldsymbol{x} \leq \boldsymbol{b}_1, \boldsymbol{A}_2 \boldsymbol{x} = \boldsymbol{b}_2, \boldsymbol{x} \geq \boldsymbol{0}\}, \end{aligned} \tag{2.30}$$

for which X is a set of feasible solutions to the (primal) problem (2.30). Later, $X^* \subset X$ will be defined as the set of optimal primal solutions to (2.30). In this problem, the total number of constraints is $m = m_1 + m_2$, and $\boldsymbol{A}_1 \in \mathbb{R}^{m_1 \times n}$, $\boldsymbol{A}_2 \in \mathbb{R}^{m_2 \times n}$, with $\boldsymbol{b}_1$ and $\boldsymbol{b}_2$ also of compatible dimensions. Alternately, the notation that $\boldsymbol{A}^T = \begin{bmatrix} \boldsymbol{A}_1^T \boldsymbol{A}_2^T \end{bmatrix}$, and $\boldsymbol{b} = \begin{pmatrix} \boldsymbol{b}_1 \boldsymbol{b}_2 \end{pmatrix}$ will be adopted.

Let a feasible problem of the form (2.30) be a problem for which $X \neq \emptyset$ and $F(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ is not infinity.

From the primal problem (2.30), the Lagrangian function can be written as

$$\begin{aligned} L_P(\boldsymbol{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = \boldsymbol{c}^T\boldsymbol{x} - \boldsymbol{\lambda}_1^T(\boldsymbol{A}_1\boldsymbol{x} - \boldsymbol{b}_1) \\ - \boldsymbol{\lambda}_2^T(\boldsymbol{A}_2\boldsymbol{x} - \boldsymbol{b}_2) + \boldsymbol{\mu}^T\boldsymbol{x}, \end{aligned} \tag{2.31}$$

where $\boldsymbol{\lambda} = \begin{pmatrix} \boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_2 \end{pmatrix}$ and $\boldsymbol{\mu}$ are lagrange multipliers. The Karush-Kuhn-Tucker (KKT) conditions that are necessary for optimal primal ($\boldsymbol{x}^*$) and dual ($\boldsymbol{\lambda}^*$) variables are

$$\begin{aligned} &\boldsymbol{x}^* \in X \\ &\boldsymbol{\lambda}_1^{*T}(\boldsymbol{A}_1\boldsymbol{x}^* - \boldsymbol{b}_1) = 0, \\ &\boldsymbol{x}^{*T}\left(\boldsymbol{c} - \boldsymbol{A}^T\boldsymbol{\lambda}^*\right) = 0, \text{ and} \\ &\boldsymbol{A}^T\boldsymbol{\lambda}^* \geq \boldsymbol{c},\ \boldsymbol{\lambda}_1^* \geq \boldsymbol{0}, \end{aligned}$$

where the i^{th} constraint is said to be active if $\lambda_i \neq 0$. This corresponds an inequality being an equality at the solution.

To state the dual program to (2.30), we first set $\boldsymbol{y} = \begin{pmatrix} \boldsymbol{y}_1 & \boldsymbol{y}_2 \end{pmatrix}$. The dual program is then given by

$$\begin{aligned} G(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) &= \min_{\boldsymbol{y} \in Y} \boldsymbol{y}^T\boldsymbol{b} \\ \text{with: } Y &= \left\{\boldsymbol{y}_1 \in \mathbb{R}^{m_1}, \boldsymbol{y}_2 \in \mathbb{R}^{m_2} | \boldsymbol{A}^T\boldsymbol{y} \geq \boldsymbol{c},\ \boldsymbol{y}_1 \geq 0\right\}, \end{aligned} \tag{2.33}$$

with its own Lagrangian function

$$L_D(\boldsymbol{y}, \boldsymbol{\pi}, \boldsymbol{\eta}) = \boldsymbol{y}^T\boldsymbol{b} - \boldsymbol{\pi}^T\left(\boldsymbol{A}^T\boldsymbol{y} - \boldsymbol{c}\right) - \boldsymbol{\eta}^T\boldsymbol{y}, \tag{2.34}$$

in which $\boldsymbol{\pi}$ and $\boldsymbol{\eta}$ are the Lagrange multipliers of the dual problem. From (2.34), we have the KKT conditions for the dual problem

$$
\begin{aligned}
&\boldsymbol{y}^* \in Y, \\
&\boldsymbol{\pi}^{*T}\left(\boldsymbol{A}^T\boldsymbol{y}^* - \boldsymbol{c}\right) = 0, \\
&\boldsymbol{y}^{*T}\left(\boldsymbol{b} - \boldsymbol{A}\boldsymbol{\pi}^*\right) = 0, \text{ and} \\
&\boldsymbol{A}_1\boldsymbol{\pi}^* \leq \boldsymbol{b}_1,\ \boldsymbol{A}_2\boldsymbol{\pi}^* = \boldsymbol{b}_2,\ \boldsymbol{\pi}^* \geq \boldsymbol{0}.
\end{aligned}
$$

From the KKT conditions for the primal and dual problem, we see that if a solution exists to (2.30), then a solution also exists to (2.33), and that at the solution, $\boldsymbol{y}^* = \boldsymbol{\lambda}^*$, $\boldsymbol{x}^* = \boldsymbol{\pi}^*$, and $F(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) = G(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ (i.e., $\boldsymbol{c}^T\boldsymbol{x}^* = \boldsymbol{y}^{*T}\boldsymbol{b}$). Furthermore, equality constraints in the primal problem lead to unrestricted variables in the dual problem, and inequality constraints in the primal problem lead to restrictions on the sign of the dual variables associated with those constraints. This is important, because the SSH LP is not in standard form as all primal variables are unrestricted in sign. The primal problem of the SSH LP is of the form

$$
\begin{aligned}
F(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) &= \max_{\boldsymbol{x} \in X} \boldsymbol{c}^T\boldsymbol{x} \\
&\text{with: } X = \left\{\boldsymbol{x} \in \mathbb{R}^n | \boldsymbol{A}_1\boldsymbol{x} \leq \boldsymbol{b}_1, \boldsymbol{A}_2\boldsymbol{x} = \boldsymbol{b}_2\right\},
\end{aligned}
\tag{2.36}
$$

and therefore the dual problem has the form

$$
\begin{aligned}
G(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) &= \min_{\boldsymbol{y} \in Y} \boldsymbol{y}^T\boldsymbol{b} \\
&\text{with: } Y = \left\{\boldsymbol{y}_1 \in \mathbb{R}^{m_1}, \boldsymbol{y}_2 \in \mathbb{R}^{m_2} | \boldsymbol{A}^T\boldsymbol{y} = \boldsymbol{c},\ \boldsymbol{y}_1 \geq 0\right\}.
\end{aligned}
\tag{2.37}
$$

In many applications either the primal solution vector, the dual solution vector, or both, are not unique. Primal degeneracy (multiple dual solutions) is quite common in applications where the number of constraints is larger than the number of variables. In particular, for problems with weakly redundant constraints active at the primal optimum, $\boldsymbol{x}^*$, there are multiple dual solutions because there is not a unique set of constraints that could be active at the solution (c.f. Akgul [1], Dantzig and Thapa [19], Eishelt and Sandblom [21], Gal [28]). Dual degeneracy (multiple primal solutions) occurs when the dual problem has weakly redundant constraints and in particular when there are linearly dependent columns in $\boldsymbol{A}$. For these cases, let us denote the sets of optimal solutions X^* and Y^*:

$$X^* = \left\{ \boldsymbol{x} \in X | \boldsymbol{c}^T \boldsymbol{x} = F(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) = G(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) \right\} \tag{2.38a}$$

$$Y^* = \left\{ \boldsymbol{y} \in Y | \boldsymbol{y}^T \boldsymbol{b} = G(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) = F(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) \right\}. \tag{2.38b}$$

Again, due to the nature of the constraints, we know that the primal and dual optimal solution sets are polyhedral in nature. Thus, if all vertices of these polyhedral sets are known, we may express any optimal solution vector as the convex combination of these vertices, according to equation (2.20).

These ideas have been explored in great detail, primarily in the areas of economics and operations research, where the primal and dual solutions have useful interpretations, c.f. Aucamp and Steinberg [5], Lin [55], as wella as[1, 19, 21, 28]. For instance, in resource allocation when there is a unique primal solution that is degenerate, the minimum and maximum values of $y_i \in Y^*$ represent the highest buying price and lowest selling price of a given resource. In a convex analysis sense, it was shown by [1] that if we allow F to be a function of $\boldsymbol{b}$, keeping $\boldsymbol{A}$ and $\boldsymbol{c}$ fixed, then $F(\boldsymbol{b})$ is a non-decreasing piecewise linear concave function and the set of subgradients $\partial F(\boldsymbol{b})$ of F at $\boldsymbol{b}$ is given by $\partial F(\boldsymbol{b}) = Y^*$.

2.5.2 Linear Program Solution Strategies

In this section, the solution to the SSH linear program based on the primal simplex algorithm (c.f. [19]) will be developed. The structure of this solution is particularly useful in evaluating the subgradient of the SSH program in cases that have not been treated in the literature. However, in order to exploit this algorithm, problems must be represented in extended standard form, and problems like the SSH LP that are in non-standard form (2.36) must first undergo a change of variables to be represented in standard form.

2.5.2.1 Extended Standard Form

The extended standard form of the primal program (2.30) is given by

$$\begin{aligned} F^{ext}(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c}) = \max_{(\boldsymbol{x}, \boldsymbol{x}_a, \boldsymbol{x}_s) \in X} & \begin{pmatrix} \boldsymbol{c} & \boldsymbol{c}_s & \boldsymbol{c}_a \end{pmatrix}^T \begin{pmatrix} \boldsymbol{x} & \boldsymbol{x}_s & \boldsymbol{x}_a \end{pmatrix} \\ \text{with: } X = \{ & \boldsymbol{x} \in \mathbb{R}^n, \boldsymbol{x}_s \in \mathbb{R}^{m_1}, \boldsymbol{x}_a \in \mathbb{R}^{m_2} | \\ & [\boldsymbol{A} \ \boldsymbol{I}] \begin{pmatrix} \boldsymbol{x} & \boldsymbol{x}_s & \boldsymbol{x}_a \end{pmatrix} = \boldsymbol{b}, \begin{pmatrix} \boldsymbol{x} & \boldsymbol{x}_s \end{pmatrix} \geq \boldsymbol{0}, \boldsymbol{x}_a = \boldsymbol{0} \Big\}, \end{aligned} \tag{2.39}$$

to which the dual problem (2.33) is also associated, and therefore (by the KKT conditions) the sets of primal and dual solution vectors to F and F^{ext} coincide, as does the value of the maximum. In the extended problem, $\boldsymbol{x}_s$ and $\boldsymbol{x}_a$ are known respectively as the slack and artificial variables, and $\boldsymbol{c}_s = \mathbf{0}$ (c.f. [19, 21]). In practice, the constraint that $\boldsymbol{x}_a = \mathbf{0}$ is never explicitly enforced. Rather, $\boldsymbol{x}_a$ are set to zero by heavily penalizing violations of this constraint and setting $\boldsymbol{c}_a = -M\mathbf{1}$ (with $M \in \mathbb{R} \gg 0$ and $\mathbf{1}$ being a vector in $\mathbb{R}^n$ with ones in all entries) or by a two phase method in which the first phase attempts to eliminate the equality constraints from the system.

2.5.2.2 Variables Unrestricted in Sign

In our problem of interest, (2.29), all primal variables are unrestricted in sign. This differs from the standard form of a linear program (2.30). In order to convert our problem with all variables $\boldsymbol{x}$ unrestricted in sign to this form, we make the change of variables

$$\boldsymbol{x} = \bar{\boldsymbol{x}} - \mathbf{1}v. \tag{2.40}$$

Note that $v = \max\{0, -\boldsymbol{x}\}$, although this last relationship need not be explicitly enforced [76]. The $\bar{(\cdot)}$ values are called the positive part of their respective variable. A related problem for $\tilde{\boldsymbol{x}} = \begin{pmatrix} \bar{\boldsymbol{x}} & v \end{pmatrix} \geq \mathbf{0} \in \mathbb{R}^{n+1}$ may then be solved, and the solution to $\boldsymbol{x}$ reconstructed from (2.40). Details can be found in [76], however we note here that this change of variables corresponds to embedding the original constraint polyhedron in $\mathbb{R}^n$ into $\mathbb{R}^{n+1}$, and the extreme points of the problem in $\mathbb{R}^n$ are the endpoints of semi-infinite rays in $\mathbb{R}^{n+1}$.

To execute this change of variables, an additional column is added to $\boldsymbol{A}$ and row to $\boldsymbol{c}$. Let us denote the augmented constraint matrix $\tilde{\boldsymbol{A}} = [\boldsymbol{A} \;\; -\boldsymbol{A}\mathbf{1}] \in \mathbb{R}^{m \times n+1}$, and the augmented objective coefficients $\tilde{\boldsymbol{c}} = \begin{pmatrix} \boldsymbol{c} & -\boldsymbol{c}^T\mathbf{1} \end{pmatrix} \in \mathbb{R}^{n+1}$. The new primal problem reads

$$\begin{aligned} F(\tilde{\boldsymbol{A}}, \boldsymbol{b}, \tilde{\boldsymbol{c}}) &= \max_{\tilde{\boldsymbol{x}} \in \tilde{X}} \tilde{\boldsymbol{c}}^T \tilde{\boldsymbol{x}} \\ \text{with: } \tilde{X} &= \left\{ \tilde{\boldsymbol{x}} \in \mathbb{R}^{n+1} | \tilde{\boldsymbol{A}}_1 \tilde{\boldsymbol{x}} \leq \boldsymbol{b}_1, \tilde{\boldsymbol{A}}_2 \tilde{\boldsymbol{x}} = \boldsymbol{b}_2, \tilde{\boldsymbol{x}} \geq \mathbf{0} \right\}, \end{aligned} \tag{2.41}$$

where the feasible region of the primal problem that has undergone this change of variables as called $\tilde{X}$. Through simple algebraic manipulation of the KKT conditions, it can readily be shown that the dual problem associated to (2.41) is not affected by the change of primal variables, and has the form of (2.37), so that $\tilde{Y} = Y$.

2.5.3 Solutions to the Primal Simplex Algorithm

Now, attention is given to solving problems of the form (2.39) by means of the primal simplex algorithm. The primal simplex algorithm works by exploring the extreme points of the constraint set in an organized way until a maximum is found. Each of these points is known as a basic feasible solution to the system of linear equations $[\boldsymbol{A}\ \boldsymbol{I}]\begin{pmatrix}\boldsymbol{x}\ \boldsymbol{x}_s\ \boldsymbol{x}_a\end{pmatrix} = \boldsymbol{b}$, (where $\boldsymbol{I}$ denotes the identity matrix), i.e., a feasible vertex in X. At each iteration, a neighboring vertex is visited so that the value of the objective is increased, and so on until the maximum is found [19, 21]. This corresponds to subdividing the primal variables $\begin{pmatrix}\boldsymbol{x}\ \boldsymbol{x}_s\ \boldsymbol{x}_a\end{pmatrix}$ into basic and non-basic variables, and exchanging one basic for one non-basic variable at each step in the primal simplex algorithm.

Let the columns of $[\boldsymbol{A}\ \boldsymbol{I}]$ and the rows of $\begin{pmatrix}\boldsymbol{x}\ \boldsymbol{x}_s\ \boldsymbol{x}_a\end{pmatrix}$ and $\begin{pmatrix}\boldsymbol{c}\ \boldsymbol{c}_s\ \boldsymbol{c}_a\end{pmatrix}$ be indexed by the index set $J = \{1, \ldots, n+1,\ \ n+2, \ldots,\ n+m+1\}$. Further, partition J into the index sets B and V, with $|B| = m$, $|V| = n$ and $V = J \setminus B$. Let $\boldsymbol{B}$ be the submatrix of $[\boldsymbol{A}\ \boldsymbol{I}]$ with columns in B, and $\boldsymbol{V}$ be the analogous submatrix with columns in V. Likewise, partition the rows of $\begin{pmatrix}\boldsymbol{x}\ \boldsymbol{x}_s\ \boldsymbol{x}_a\end{pmatrix}$ and $\begin{pmatrix}\boldsymbol{c}\ \boldsymbol{c}_s\ \boldsymbol{c}_a\end{pmatrix}$ accordingly into $\boldsymbol{x}_B$ and $\boldsymbol{x}_V$, and $\boldsymbol{c}_B$ and $\boldsymbol{c}_V$.

Further, introduce the index sets $R = \{1, \ldots, n\}$ and $S = \{n+1, \ldots, n+m\}$ where R corresponds to the indices (columns of $[\boldsymbol{A}_v\ \boldsymbol{I}]$, rows of $\begin{pmatrix}\boldsymbol{x}\ \boldsymbol{x}_s\ \boldsymbol{x}_a\end{pmatrix}$ and $\begin{pmatrix}\boldsymbol{c}\ \boldsymbol{c}_s\ \boldsymbol{c}_a\end{pmatrix}$) of the original variables, and S to the slack and artificial variables. The index set S can be partitioned into S_1 and S_2, corresponding to the slack and artificial variables, respectively.

If $F(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ has a finite maximum, the following inequality holds at the maximum

$$(\boldsymbol{c}_B^T \boldsymbol{B}^{-1}\boldsymbol{V} - \boldsymbol{c}_V^T)_{V \cap S_1} \geq 0, \tag{2.42}$$

where the rows corresponding the S_2 are either trivially $\gg 0$ if the big-M method has been used, or not considered if a two-phase method has been used. For a primal solution vector $\boldsymbol{x}^* = \begin{pmatrix}\boldsymbol{x}_B\ \boldsymbol{x}_V\end{pmatrix}$, $\boldsymbol{x}_B$ has the form

$$\boldsymbol{x}_B = \boldsymbol{B}^{-1}(\boldsymbol{b} - \boldsymbol{V}\boldsymbol{x}_V) \geq 0. \tag{2.43}$$

If all entries in (2.42) are strictly greater than zero, then the primal solution vector is unique, and $\boldsymbol{x}_V = \boldsymbol{0}$. If the solution is not unique, then let $Q \subset V$ be the index set for which $\boldsymbol{c}_B^T \boldsymbol{B}^{-1}\boldsymbol{V} - \boldsymbol{c}_V^T = 0$, and $\overline{Q}$ be $V \setminus Q$. The full primal solution set is given by Best [8],[1]:

$$X^* = \left\{ \left[\boldsymbol{x}_B \ \boldsymbol{x}_Q \ \boldsymbol{x}_{\overline{Q}} \right] | \boldsymbol{x}_B = \boldsymbol{B}^{-1}(\boldsymbol{b} - \boldsymbol{V}_Q \boldsymbol{x}_Q), \right. \\ \left. \boldsymbol{x}_B \geq \boldsymbol{0}, \boldsymbol{x}_Q \geq \boldsymbol{0}, \boldsymbol{x}_{\overline{Q}} = \boldsymbol{0} \right\}. \tag{2.44}$$

To find the dual solution, the row vector $\overline{\boldsymbol{c}} \in \mathbb{R}^{n+m}$ is used following

$$\overline{\boldsymbol{c}} = \left(\boldsymbol{0}_B \ (\boldsymbol{c}_B^T \boldsymbol{B}^{-1} \boldsymbol{V} - \boldsymbol{c}_V^T)_V \right), \tag{2.45}$$

where $\boldsymbol{0}_B$ is the $m \times 1$ zero vector indexed by B, and all previously introduced index sets, as appropriate, are embedded in $\overline{\boldsymbol{c}}$. The values in $\overline{\boldsymbol{c}}$ are commonly referred to as the 'reduced cost coefficients'. From this a solution to the dual problem can be found as

$$\boldsymbol{y}^* = \overline{\boldsymbol{c}}_S. \tag{2.46}$$

If the big-M method is used, then at the solution any remaining factors of M are subtracted off of $\overline{\boldsymbol{c}}$ to find the dual solution of the original problem. If $\boldsymbol{x}_B > \boldsymbol{0}$ for all components, then the optimal dual vector $\boldsymbol{y}^*$ given in (2.46) is unique, i.e., the primal solution is non-degenerate. However, if any component of $\boldsymbol{x}_B$ is zero, then it is possible to alter the index set B, and thus alter the dual solution vector, without changing the primal solution vector or the optimal value of $F(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$. Let us denote the index set $T \subset B$ as the rows for which $\boldsymbol{x}_B = 0$. We subsequently use the notation $\{B\}$ to denote the set of unique index sets that may represent the solution to (2.39). Y^* is of the form [1, 8]

$$Y^* = \left\{ \boldsymbol{y} | \boldsymbol{y}^T = \overline{\boldsymbol{c}}_S^T + \boldsymbol{t}^T (\boldsymbol{B}^{-1} [\boldsymbol{B} \ \boldsymbol{V}])_{T,S}, \right. \\ \left. \overline{\boldsymbol{c}}_{R \cup S_1}^T + \boldsymbol{t}^T (\boldsymbol{B}^{-1} [\boldsymbol{B} \ \boldsymbol{V}])_{T, R \cup S_1} \geq 0, \ \boldsymbol{t} \in \mathbb{R}^{|T|} \right\}, \tag{2.47}$$

where the notation $(\cdot)_{I,J}$ refers to the submatrix with rows in the index set I and columns in the index set J, and $\boldsymbol{t} \in \mathbb{R}^{|T|}$ is a vector of parameters. Obviously, if $T = \emptyset$, then $Y^* = \{\boldsymbol{y}^*\}$.

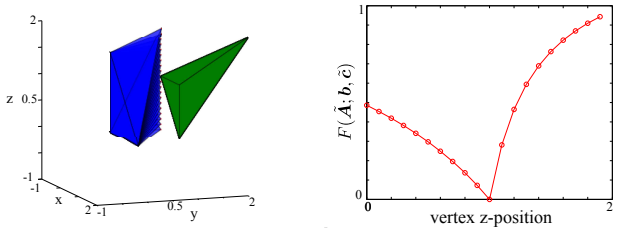
Finally, the unique value of the maximum is given by

$$\begin{aligned} F(\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}) &= \boldsymbol{c}_B^T\boldsymbol{x}_B + \boldsymbol{c}_V^T\boldsymbol{x}_V \\ &= \boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{b} - (\boldsymbol{c}_B^T\boldsymbol{B}^{-1}\boldsymbol{V} - \boldsymbol{c}_V^T)\boldsymbol{x}_V. \end{aligned} \tag{2.48}$$

2.6 Nonsmooth Analysis

Various authors, primarily in the fields of economics and operations research, have explored what is known in the literature as the 'sensitivity' of a linear program to its data $(\boldsymbol{A},\boldsymbol{b},\boldsymbol{c})$ beginning in the 1950's with the Dantzig's inception of the topic. That is, if F is a smooth function of $\boldsymbol{A}$, $\boldsymbol{b}$, and $\boldsymbol{c}$, the values of interest in sensitivity analysis are $\frac{\partial F}{\partial \boldsymbol{b}}$, $\frac{\partial F}{\partial \boldsymbol{c}}$, and $\frac{\partial F}{\partial \boldsymbol{A}}$. The former two are well understood, in particular for a program in standard form. However, the vast majority of economic analyses take the entries in $\boldsymbol{A}$ to be fixed. Those analyses of the sensitivity of a linear program to the coefficients in the contraint matrix that do exist make use of several regularity conditions (e.g. the problems treated are in standard form, with no linearly dependent columns or rows) that in general are not present for the SSH LP [26, 27, 59].

For the SSH linear program, we are interested in how $F(\tilde{\boldsymbol{A}};\boldsymbol{b},\tilde{\boldsymbol{c}})$ changes with respect to the entries in the original matrix $\boldsymbol{A}$. An example of this dependence for an SSH linear program (2.29) that has been put into extended form and undergone the change of variables in Section 2.5.2.2 is shown in Figure 2.10. In this example, the $z-$coordinate of one tetrahedron is varied and the vector $\hat{\boldsymbol{\beta}}$ is held constant.



(a) (b)

Figure 2.10: The dependence of the maximum function $F(\tilde{\boldsymbol{A}};\boldsymbol{b},\tilde{\boldsymbol{c}})$ on the data in the constraint matrix. One entry in the constraint matrix $\boldsymbol{A}$ is varied by increasing the $z-$coordinate of a vertex of the tetrahedron on the left.

In general, the maximum function is neither a smooth nor a convex function of entries in $\boldsymbol{A}$. However, we have shown that the SSH linear program is always bounded and feasible for an appropriate choice of $\hat{\boldsymbol{\beta}}$, and it follows directly from the structure of an LP solution described in Section 2.5.3 that for problems of the form of (2.29), $F(\tilde{\boldsymbol{A}};\boldsymbol{b},\boldsymbol{c})$ is Lipschitz over $A_{ij} \in \mathbb{R}$. With this in mind, we take

a general non-smooth analysis approach to understand the change in the solution with respect to the entries in $\boldsymbol{A}$ in the original matrix for problems of the type (2.29) with the information at hand from a problem solved in a higher dimension using $\tilde{\boldsymbol{A}}$.

2.6.1 Generalized Differential

To begin, we introduce the generalized directional derivative from Clarke [16]. First, we let f be a Lipschitz mapping from $f : X \to \mathbb{R}$, where X has an associated dual space, Y. We also define the duality pairing $\langle \cdot, \cdot \rangle$, between X and Y. Clarke's generalized directional derivative is then given by

$$f^{\circ}(\boldsymbol{x}; \boldsymbol{\lambda}) = \limsup_{t \to 0^{+},\ \boldsymbol{x}' \to \boldsymbol{x}} \frac{f(\boldsymbol{x}' + t\boldsymbol{\lambda}) - f(\boldsymbol{x}')}{t}, \tag{2.49}$$

where $\boldsymbol{x}', \boldsymbol{\lambda} \in X$ and t is a positive scalar. Clarke's generalized subdifferential (or generalized gradient, which for simplicity we alternately refer to as the subgradient) of a Lipschitz function f at $\boldsymbol{x}$ is the subset

$$\partial f = \{\boldsymbol{y} \in Y | f^{\circ}(\boldsymbol{x}; \boldsymbol{\lambda}) \geq \langle \boldsymbol{y}, \boldsymbol{\lambda} \rangle\,,\ \forall \boldsymbol{\lambda} \in X\}\,, \tag{2.50}$$

for $\boldsymbol{y} \in Y$. The generalized directional derivative can be recovered from the generalized gradient as

$$f^{\circ}(\boldsymbol{x}; \boldsymbol{\lambda}) = \max\{\langle \boldsymbol{y}, \boldsymbol{\lambda} \rangle | \boldsymbol{y} \in \partial f\}\,. \tag{2.51}$$

In the following sections, we develop an expression for the generalized directional derivative and use this result to recover an expression for the generalized gradient.

2.6.2 Generalized Directional Derivative of $F(\boldsymbol{A})$

We first develop an expression for $F^{\circ}(\boldsymbol{A}; \boldsymbol{\Lambda}, \boldsymbol{b}, \boldsymbol{c})$, which we shorten to $F^{\circ}(\boldsymbol{A}; \boldsymbol{\Lambda})$, in the case where both (2.30) and (2.33) may be degenerate. Following Freund [26, 27], Williams [87] and Mills [59], we adopt a perturbation matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{m \times (n+m)}$, with a one in the entry of interest, and zeros in all other entries.

$$\boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{0}^n & \mathbf{0}^m \\ \vdots & \vdots \\ \mathbf{0}^n & \mathbf{0}^m \\ e_j & \mathbf{0}^m \\ \mathbf{0}^n & \mathbf{0}^m \\ \vdots & \vdots \\ \mathbf{0}^n & \mathbf{0}^m \end{bmatrix} \in \mathbb{R}^{m\times(n+m)}. \tag{2.52}$$

In (2.52), $\mathbf{0}^k \in \mathbb{R}^k$ ($k = n$ or m) is a row vector with zeros in all entries, and $e_j \in \mathbb{R}^n$ is a row vector with a one in the entry of interest, and zeros in all other entries. Let us call $\overline{\boldsymbol{A}} = \boldsymbol{A} + t\boldsymbol{\Lambda}$. The solution to $F(\boldsymbol{A} + t\boldsymbol{\Lambda}) = F(\overline{\boldsymbol{A}})$ is given by

$$F(\overline{\boldsymbol{A}}) = \boldsymbol{c}_B^T \overline{\boldsymbol{B}}^{-1}\boldsymbol{b} - (\boldsymbol{c}_B^T \overline{\boldsymbol{B}}^{-1}\overline{\boldsymbol{V}} - \boldsymbol{c}_V^T)\boldsymbol{x}_V. \tag{2.53}$$

We assume that the perturbation leaves $\overline{\boldsymbol{B}}$ invertible, which is certainly reasonable as $t \to 0^+$. In this expression, the index sets B and V now subdivide the perturbed matrix $\left(\overline{\boldsymbol{A}}\,\boldsymbol{I}\right)$ into $\overline{\boldsymbol{B}}$ and $\overline{\boldsymbol{V}}$, and $\boldsymbol{\Lambda}$ into $\boldsymbol{\Lambda}_B$ and $\boldsymbol{\Lambda}_V$. We have the relationship

$$\overline{\boldsymbol{B}}\,\overline{\boldsymbol{B}}^{-1} = (\boldsymbol{B} + t\boldsymbol{\Lambda}_B)(\boldsymbol{B} + t\boldsymbol{\Lambda}_B)^{-1} = \boldsymbol{I}, \tag{2.54}$$

which can be premultiplied by $\boldsymbol{B}^{-1}$ and rearranged such that

$$\overline{\boldsymbol{B}}^{-1} = \boldsymbol{B}^{-1} - t(\boldsymbol{B}^{-1}\boldsymbol{\Lambda}_B)\overline{\boldsymbol{B}}^{-1}, \tag{2.55}$$

from which we can derive an expression for $\overline{\boldsymbol{B}}^{-1}$ as an infinite series

$$\overline{\boldsymbol{B}}^{-1} = \sum_{i=0}^{\infty} t^i(-\boldsymbol{B}^{-1}\boldsymbol{\Lambda}_B)^i\boldsymbol{B}^{-1}. \tag{2.56}$$

This series converges for all $|t| \leq \epsilon = (k\left\|-\boldsymbol{B}^{-1}\boldsymbol{\Lambda}_B\right\|)^{-1}$ for some $k > 0$. Using $\overline{\boldsymbol{H}} = \boldsymbol{H} + t\boldsymbol{\Lambda}$, this allows us to develop an explicit expression for the generalized directional derivative of F at $\boldsymbol{A}$ as

$$
\begin{aligned}
F^{\circ}(\boldsymbol{A};\boldsymbol{\Lambda}) &= \limsup_{t\to 0^{+},\ \boldsymbol{H}\to\boldsymbol{A}} \frac{1}{t}\left(F(\overline{\boldsymbol{H}}) - F(\boldsymbol{H})\right) \\
&= \limsup_{\substack{t\to 0^{+},\ B\in\{B\} \\ \boldsymbol{x}\in X^{*}}} \frac{1}{t}\Bigg\{ \boldsymbol{c}_B^T \left[\sum_{i=0}^{\infty} t^i(-\boldsymbol{B}^{-1}\boldsymbol{\Lambda}_B)^i\boldsymbol{B}^{-1}\right]\boldsymbol{b} \\
&\quad -\Bigg(\boldsymbol{c}_B^T \left[\sum_{i=0}^{\infty} t^i(-\boldsymbol{B}^{-1}\boldsymbol{\Lambda}_B)^i\boldsymbol{B}^{-1}\right](\boldsymbol{V}+t\boldsymbol{\Lambda}_V) \\
&\quad -\boldsymbol{c}_V^T\Bigg)\boldsymbol{x}_V - F(\boldsymbol{A})\Bigg\} \\
&= \sup_{B\in\{B\},\ \boldsymbol{x}\in X^{*}} -\boldsymbol{c}_B^T\boldsymbol{B}^{-1}\left(\boldsymbol{\Lambda}_B\boldsymbol{B}^{-1}(\boldsymbol{b}-\boldsymbol{V}\boldsymbol{x}_V)\right. \\
&\quad \left. +\boldsymbol{\Lambda}_V\boldsymbol{x}_V\right),
\end{aligned}
$$

which is equivalent to

$$
\begin{aligned}
F^{\circ}(\boldsymbol{A};\boldsymbol{\Lambda}) &= \sup_{B\in\{B\},\ \boldsymbol{x}\in X^{*}} -\boldsymbol{c}_B^T\boldsymbol{B}^{-1}(\boldsymbol{\Lambda}_B\boldsymbol{x}_B + \boldsymbol{\Lambda}_V\boldsymbol{x}_V) \\
&= \sup_{B\in\{B\},\ \boldsymbol{x}\in X^{*}} -\boldsymbol{c}_B^T\boldsymbol{B}^{-1}(\boldsymbol{\Lambda}\boldsymbol{x}).
\end{aligned}
\tag{2.57}
$$

In the preceeding sets of equations, $\{B\}$ denotes the set of possible index sets B corresponding to basic feasible solutions, i.e. solutions to the problem which lie at vertices of the feasible primal and dual polyhedrons so that $\boldsymbol{B}$ is square and has full rank. From the KKT conditions, for a given basic feasible solution, we can find a complementary dual solution which satisfies:

$$
\boldsymbol{y}^{*T}\boldsymbol{B} = \boldsymbol{c}_B. \tag{2.58}
$$

This allows us to write (2.57) as:

$$
F^{\circ}(\boldsymbol{A};\boldsymbol{\Lambda}) = \sup_{\boldsymbol{y}\in Y_B^{*},\ \boldsymbol{x}\in X^{*}} -\boldsymbol{y}^T\boldsymbol{\Lambda}\boldsymbol{x}, \tag{2.59}
$$

where $Y_B^{*}\subset Y^{*}$ is the set of basic feasible dual solutions. However, because of the affine nature of X^{*} and Y^{*}, (2.59) is equivalent to:

$$F^{\circ}(\boldsymbol{A};\boldsymbol{\Lambda}) = \sup_{\boldsymbol{y}\in Y^*,\ \boldsymbol{x}\in X^*} -\boldsymbol{y}^T\boldsymbol{\Lambda}\boldsymbol{x}. \tag{2.60}$$

Note that in (2.60), we have not assumed anything about the sign of $\boldsymbol{x}$ or whether (or not) the problem is primal degenerate, dual degenerate, or both.

2.6.3 $F^{\circ}(\boldsymbol{A};\boldsymbol{\Lambda})$ for Unrestricted Variables

We now use the expression in equation (2.60) to evaluate the generalized directional derivative in the case of the specific change in variables given in Section 2.5.2.2. For this problem, the structure of $\tilde{X}^*$ for the augmented problem $F(\tilde{\boldsymbol{A}}, \boldsymbol{b}, \tilde{\boldsymbol{c}})$ is such that we fully expect to find multiple solutions $\boldsymbol{x}^* \in X^*$ (c.f. [76]), which suggests that it is possible for the generalized directional derivative $F^{\circ}(\tilde{\boldsymbol{A}};\boldsymbol{\Lambda})$ to take on infinite values. Luckily, we are not interested in $F^{\circ}(\tilde{\boldsymbol{A}};\boldsymbol{\Lambda})$ at face value, but rather in its projection onto a lower dimensional space. This may be accomplished by introducing a linearly dependent column into the perturbation matrix, which reflects the relationship $\boldsymbol{x} = \bar{\boldsymbol{x}} - \mathbf{1}v$.

In order to find the generalized directional derivative with respect to the data of the original problem, $\boldsymbol{\Lambda}$ takes on the form $\tilde{\boldsymbol{\Lambda}}$

$$\tilde{\boldsymbol{\Lambda}} = \begin{bmatrix} \mathbf{0}^n & 0 & \mathbf{0}^m \\ \vdots & \vdots & \vdots \\ \mathbf{0}^n & 0 & \mathbf{0}^m \\ e_j & -1 & \mathbf{0}^m \\ \mathbf{0}^n & 0 & \mathbf{0}^m \\ \vdots & \vdots & \vdots \\ \mathbf{0}^n & 0 & \mathbf{0}^m \end{bmatrix} \in \mathbb{R}^{m\times(n+m+1)}. \tag{2.61}$$

The generalized directional derivative with respect to the original entries in $\boldsymbol{A}$ is then

$$F^{\circ}(\tilde{\boldsymbol{A}};\tilde{\boldsymbol{\Lambda}}) = \sup_{\tilde{\boldsymbol{y}}\in\tilde{Y}^*,\ \tilde{\boldsymbol{x}}\in\tilde{X}^*} -\tilde{\boldsymbol{y}}^T\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{x}}. \tag{2.62}$$

However, we know that $Y = \tilde{Y}^*$ and that

$$\tilde{\mathbf{\Lambda}}\tilde{\boldsymbol{x}} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \bar{x}_j - v \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ x_j \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \mathbf{\Lambda}\boldsymbol{x}. \tag{2.63}$$

From these relationships, we have that:

$$F^{\circ}(\tilde{\boldsymbol{A}}; \tilde{\mathbf{\Lambda}}) = F^{\circ}(\boldsymbol{A}; \mathbf{\Lambda}) = \sup_{\boldsymbol{y} \in Y^*,\ \boldsymbol{x} \in X^*} -\boldsymbol{y}^T \mathbf{\Lambda} \boldsymbol{x}. \tag{2.64}$$

From equation (2.64), we can infer a simple expression for the generalized gradient as:

$$\partial F(\boldsymbol{A}; \mathbf{\Lambda}) = \left\{ -\boldsymbol{y}^T \mathbf{\Lambda} \boldsymbol{x} | \boldsymbol{y} \in Y^*, \boldsymbol{x} \in X^* \right\}. \tag{2.65}$$

Finally, we note that if the problem is not degenerate, then the generalized gradient is the normal gradient, with the value

$$\nabla_{\mathbf{\Lambda}} F(\boldsymbol{A}) = -\boldsymbol{y}^T \mathbf{\Lambda} \boldsymbol{x}. \tag{2.66}$$

Remark 2.7. *The SSH LP is structured so that any solution vector, which we call $\boldsymbol{\alpha}$ in (2.29), is restricted to lie in the normal cone or its reflection at some point on the boundary of each set. Furthermore, the KKT conditions state that the dual variable corresponding to the i^{th} constraint is zero if that constraint is not active at the solution, and non-zero if the constraint is active. In the SSH problem, a non-zero dual value has a one-to-one correspondence to the associated vertex being a support point for one of the optimal set of hyperplanes from the SSH test, and therefore being associated with a feature involved in a collision. Furthermore (for a non-degenerate case) the expression for the gradient reads*

$$\begin{aligned} \nabla_{\boldsymbol{x}_i} g &= \ \ y_i \boldsymbol{\alpha},\ \boldsymbol{x}_i \in \operatorname{ext} K_1 \\ \nabla_{\boldsymbol{x}_i} g &= -y_i \boldsymbol{\alpha},\ \boldsymbol{x}_i \in \operatorname{ext} K_2, \end{aligned} \tag{2.67}$$

where y_i is the dual variable associated with the i^{th} constraint. For degenerate problems, the generalized gradient has a similar structure, in that it is always in the direction $\boldsymbol{\alpha}$ or $-\boldsymbol{\alpha}$, modulo a constant non-negative factor (y_i). Therefore, not only is the subgradient non-zero for only those vertices associated with features involved in the collision, but, because we have solved for $\boldsymbol{\alpha}$ via optimization, the (generalized) gradient satisfies $\partial g \in -N_{\mathcal{A}}(\boldsymbol{x})$, where $\boldsymbol{x}$ is a feature involved in the collision.

Remark 2.8. *The convention for an interpenetration constraint function is that $g(\boldsymbol{x}) < 0$ implies an admissible configuration, and $g(\boldsymbol{x}) > 0$ implies interpenetration. This can clearly be achieved by the program in equation (2.68).*

$$\begin{aligned} g(K_1, K_2) = - \max_{\substack{\boldsymbol{\alpha}\in\mathbb{R}^n \\ a_1,a_2\in\mathbb{R}}} \; & a_1 - a_2, \;\; \textit{Subject to:} \\ & \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle - a_1 \geq 0, \;\; \boldsymbol{x} \in \operatorname{ext} K_1 \\ & \langle \boldsymbol{\alpha}, \boldsymbol{y} \rangle - a_2 \leq 0, \;\; \boldsymbol{y} \in \operatorname{ext} K_2 \\ & \left\langle \hat{\boldsymbol{\beta}}, \boldsymbol{\alpha} \right\rangle - 1 = 0 \end{aligned} \tag{2.68}$$

For this program, the generalized gradient is the same as in (2.67), multiplied by a factor of -1. In this case, $\partial g \in N_{\mathcal{A}}(\boldsymbol{x})$ for features in each body involved in the collision, as illustrated for several different collisions between two tetrahedra in three dimensions in Figure 2.12. Note that a force system $\boldsymbol{f} \in -\partial g$ will be in the direction of normal forces on the system due to contact.

2.7 Implementation

2.7.1 Algorithm Overview

In the previous sections, the focus has been on developing the SSH LP for convex polytopes from the SSH QCLP for general conves bodies, and showing that the SSH LP's subderivative may be evaluated analytically and in closed form once a solution to the SSH LP is known. The algorithm is meant as a general fine scale collision detection algorithm for polytopes, however the following pseudocode effectively summarizes the overall algorithm. The results of the algorithm are whether or not a collision has occured, and the optimal primal and dual variables, or their range in the case of degenerate problems.

Algorithm 1 SSH Collision Detection for Polytopes

```
Run coarse scale collision detection
for all Pairs of possibly intersecting polytopes, (K_1, K_2) do
   Calculate β̂, assemble equality constraint vector A_2
   Assemble inequality constraint matrix A_1 from vertex sets of bodies
   Evaluate SSH LP (Equation (2.29)) using the Simplex method
   if Subderivative information is needed then
      if Degeneracy detected then
         Follow Akgul [1] to determine α*, y* range
      end if
      Return g(K_1, K_2), α*, y* (range)
   else
      Return g(K_1, K_2)
   end if
end for
```

2.7.2 Computational Complexity

The computational complexity for the fine scale collision detection test is equivalent to that of evaluating the SSH LP, which is limited by the number of operations needed to invert a matrix of a given size as part of the solution procedure for the linear program. For each pair of bodies, this amounts to $C_{LP}(|\operatorname{ext} K_1| + |\operatorname{ext} K_2| + 1)^3$ floating point operations (flops), where $C_{LP} \approx 2$ after initialization so that the SSH LP can be warm started from the last known basis. In order to compare to the OOS test introduced in Section 2.1, it is important to recall that the OOS test requires at least the surface of the sets K_1 and K_2 to be triangulated. Using the notation $E_{n-1}(T_j)$ to denote the set of simplices of dimension $n-1$ the triangulation of the j^{th} set, the OOS test requires $C_{OOS}|E_{n-1}(T_1)||E_{n-1}(T_2)|$ flops per pair of bodies, and $C_{OOS} = 30$ for $n = 3$ and $C_{OOS} = 7$ for $n = 2$, for the test in [38]. So, if one is only interested in whether or not the bodies are penetrating, a test based on overlapping oriented simplices is the most efficient choice so long as the triangulation of the bodies does not render $|E_{n-1}(T_1)||E_{n-1}(T_2)| \approx (|\operatorname{ext} K_1| + |\operatorname{ext} K_2| + 1)^3$. However, we would suggest that for many applications the fact that the bodies in consideration need not be triangulated as well as the additional information provided by the SSH LP evaluation merits its use. Furthermore, if one is not judicious in the triangulation of the bodies, it is possible for $|E_{n-1}(T_1)||E_{n-1}(T_2)| \approx (|\operatorname{ext} K_1| + |\operatorname{ext} K_2| + 1)^3$, in which case the SSH LP approaches the efficiency of the OOS test.

2.7.3 Linear Programming Solver

Because of the small dimension of the proposed linear program (n=6, m=9 for two tetrahedra) and the known shortcomings of commercial software in adequately describing the full solution set

in the case of primal degeneracy, a basic two-phase version of the revised primal simplex method was implemented following [19, 21, 65]. Note that we did *not* implement the Big-M method, as in practice it can lead to significant conditioning problems (see [19, 21, 65] for details). Using this approach, a dual solution is immediately available at the end of the optimization. For degenerate cases, which occur quite infrequently in practice, the method of Akgul [1] is used to resolve the component-wise range of the dual and primal solution sets. Finally, after initialization, we use a 'warm start' procedure by starting the optimization at the last known optimal basis (index set B).

2.7.4 Choice of $\boldsymbol{\beta}$

All examples in this and the following chapters were run with the vector $\boldsymbol{\beta}$ set to point from the arithmetic mean of $\operatorname{ext} K_2$ to the arithmetic mean of $\operatorname{ext} K_1$,

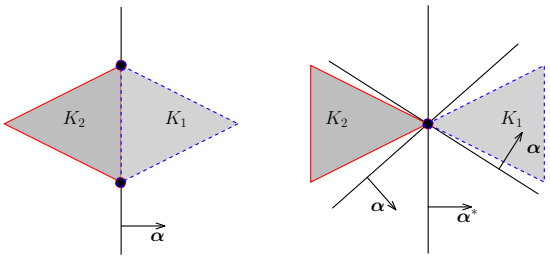
$$\boldsymbol{\beta} = \frac{\frac{1}{|\operatorname{ext} K_1|}\sum_{\boldsymbol{x}\in\operatorname{ext} K_1}\boldsymbol{x} - \frac{1}{|\operatorname{ext} K_2|}\sum_{\boldsymbol{y}\in\operatorname{ext} K_2}\boldsymbol{y}}{\left\|\frac{1}{|\operatorname{ext} K_1|}\sum_{\boldsymbol{x}\in\operatorname{ext} K_1}\boldsymbol{x} - \frac{1}{|\operatorname{ext} K_2|}\sum_{\boldsymbol{y}\in\operatorname{ext} K_2}\boldsymbol{y}\right\|}, \tag{2.69}$$

for finite element simulations, or a vector from the center of mass of K_2 to the center of mass of K_1 in rigid body simulations. This choice of β works well so long as the arithmetic means are not the same point (causing $\boldsymbol{\beta} = \mathbf{0}$), which we expect to be the case for most applications. This choice of $\boldsymbol{\beta}$ renders the SSH LP continuous over the entire domain of the simulation, because $\boldsymbol{\beta}$ is made a smooth function of the data. It is worth noting that (2.69) is not the only acceptable choice for $\boldsymbol{\beta}$. For instance, if the method of Chung and Wang [14] is used as a coarse search for collisions, then the separating vector from this approach may be used, or the optimal direction $\boldsymbol{\alpha}$ from the previous solution to the problem could also be used.

The downside of this choice for $\boldsymbol{\beta}$ is that it does induce a dependence on $\operatorname{ext} K_1$ and $\operatorname{ext} K_2$ in the equality constraint of the problem (2.29) and therefore the chain rule must be used in the evaluation of the subgradient. If (2.29) is being used directly as a contact potential and its subgradient to determine contact forces, this choice of $\boldsymbol{\beta}$ causes forces on vertices that do not bound features involved in the collision. However, the value of the dual variable associated with the equality constraint, and therefore the magnitude of these non-local forces, is exactly the value of the optimum because, at the solution, $\boldsymbol{b}^T\boldsymbol{y}^* = \boldsymbol{c}^T\boldsymbol{x}^*$ for all $\boldsymbol{x}^* \in X^*$ and $\boldsymbol{y}^* \in Y^*$, and there is only one non-zero entry in $\boldsymbol{b}$. Again, if a given simulation is set up so that any overlap of colliding bodies is small, then the relative magnitude of the non-local forces will also be small. Alternately, these forces may simply be ignored in order to approximate $N_{\mathcal{A}}(\boldsymbol{x})$ in the closest admissible configuration, where $g(\boldsymbol{x}) = 0$.

2.7.5 Subgradient Direction

In practice, degeneracies (either primal or dual) are rarely encountered in the SSH linear program. This is because primal and dual degeneracy are directly related to exact geometric symmetries in the contact configuration, which are rarely present within the tolerance of a given simulation. However, if degeneracy does occur, a single subgradient direction can be chosen to reflect these symmetries. This is illustrated through two two-dimensional examples: face-face and corner-corner contact between simplicial bodies, which are sketched in Figure 2.11.



(a) Primal degenerate problem/Multiple dual solutions

(b) Dual degenerate problem/Multiple primal solutions

Figure 2.11: Primal degenerate (face-face) and dual degenerate (corner-corner) contact configurations. For each case, the vertices associated with the dual solution are marked with a black dot, and the hyperplane(s) associated with the primal solution are shown. In Figure 2.11b, the primal solution that represents the symmetry of the contact configuration is noted as $\boldsymbol{\alpha}^*$.

The case of primal degeneracy (multiple dual solutions) is treated first. To begin, recall that primal degeneracy occurs when there are multiple weakly redundant constraints, which means that any subset of these weakly redundant constraints may be active at the solution, but the simplex algorithm terminates once one set of active constraints is found. In other words, the set of dual solutions is a bounded polyhedral set, but the simplex algorithm terminates once it reaches one of the vertices of this set. For the SSH LP, there always are weakly redundant constraints present in the case of face-face contact between bodies (see Figure 2.11a).

To begin, let us index the rows of $\boldsymbol{A}$ (columns of $\boldsymbol{A}^T$) by the index set, J_d. We may identify the set of all active constraints as $A_d \subset J_d$ as the equality constraint plus the inequality constraints associated with dual variables y_i for which $\max_{y_i \in Y^*} y_i^2 \neq 0$. Because all constraints in the dual problem to the SSH LP are equality constraints we have identified a reduced underdetermined system for the active dual variables

$$(\boldsymbol{A}_{A_d,B})^T \boldsymbol{y}^*_{A_d} = \boldsymbol{c}, \tag{2.70}$$

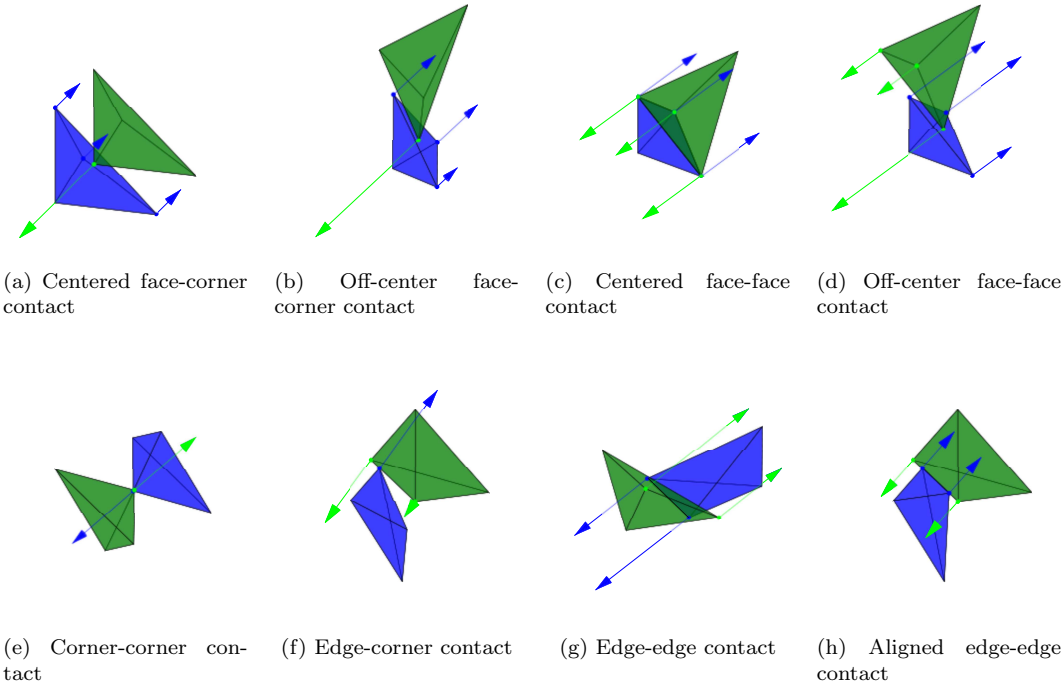
where $\boldsymbol{A}_{A_d,B}$ refers to the submatrix of $\boldsymbol{A}$ whose rows correspond to weakly redundant constraints at the solution, plus the equality constraint. The Moore-Penrose pseudo-inverse can be used to find a unique solution to this system that gives all of the non-zero entries in $\boldsymbol{y}^* \in Y^*$. This method corresponds geometrically to the orthogonal projection of $\mathbf{0}$ in the dual space onto the affine optimal solution set Y^*, and is precisely the correct operation to select $\boldsymbol{y}^* \in Y^*$ because the structure of the dual constraint set is such that symmetries in the contact configuration correspond to symmetries of the optimal set with respect to the origin of the dual space. For example, in a perfectly aligned face-face collision, the orthogonal projection of the origin of the dual space onto Y^* is equidistant from the vertices of Y^*.

The case of multiple primal solutions (dual degeneracy) for the SSH LP is easier to visualize, and corresponds in two dimensions to vertex-vertex contact between bodies, as is shown in Figure 2.11b. In this case, the set of dual constraints active for all primal solutions can be readily found. An underdetermined system for an optimal primal solution vector can be written as

$$\boldsymbol{A}_{A_d,B}\boldsymbol{x}^* = \boldsymbol{b}_{A_d}, \tag{2.71}$$

where for the SSH LP, $\boldsymbol{x}^* = (\boldsymbol{\alpha}^* \; a_1^* \; a_2^*)$. Again, this solution is the orthogonal projection of $\mathbf{0}$ in the primal space onto X^*. Perhaps a more useful physical interpretation of the minimum norm solution $\boldsymbol{x}^* \in X^*$ to an underdetermined SSH primal problem is exactly that it is the solution with the smallest Euclidean norm, which corresponds to the direction in which the smallest contact force or impulse should be applied to prevent the interpenetration of matter, subject, of course, to any bias imposed by the choice of $\boldsymbol{\beta}$.

Figure 2.12 shows several collisions of two tetrahedra in three dimensions. The subgradient with respect to each configuration variable–the vertex positions in this case–is plotted using $\boldsymbol{\beta}$ in equation (2.69), and the minimum norm solution to primal and/or dual variables in several degenerate cases. In these figures, the scale of the vectors reflects the relative magnitude of the dual variable associated with each constraint, and the direction reflects the primal solution, $\boldsymbol{\alpha}$.



(a) Centered face-corner contact

(b) Off-center face-corner contact

(c) Centered face-face contact

(d) Off-center face-face contact

(e) Corner-corner contact

(f) Edge-corner contact

(g) Edge-edge contact

(h) Aligned edge-edge contact

Figure 2.12: Several examples of collisions between two tetrahedra in 3d. For all degenerate problems, the subgradient direction shown corresponds to the minimum-norm solution to the reduced underdetermined problem. Figures 2.12a, 2.12b, and 2.12g show non-degenerate contact configurations. Figures 2.12c and 2.12d show primal degenerate problems with multiple dual solutions. Figures 2.12e and 2.12f show dual degenerate problems, and figure 2.12h is an example of a collision configuration that leads to an SSH LP that is both primal and dual degenerate.

2.8 Examples

2.8.1 Collision Type and Closest Features

The dual solution to the SSH linear program can be used directly to determine the type of collision that has taken place. As noted in the remarks at the end of the previous section, any non-zero dual variables correspond to active constraints at the solution. The number of active constraints associated with each body can be used to determine the type of contact that has taken place. This can be seen in Figure 2.12, where vertices associated with non-zero dual variables are highlighted. For example, if both bodies are tetrahedra, if there is one non-zero dual variable in the first set of constraints, and three non-zero dual variables in the second set, then a vertex of the first body is in contact with a face of the second body (see Figures 2.12a and 2.12b). If a collision has not occurred, then the closest features can be determined by the same method.

Furthermore, the closest *vertex* in K_1 to K_2 is the vertex associated with the largest dual variable.

This can be seen from the generalized derivative developed in Section 2.6; i.e., if the expression in (2.66) is examined, it is clear that if the dual variable for a given vertex constraint is zero, then small changes in that constraint do not affect the value of (2.29) and that constraint is redundant in the given configuration. Likewise, small changes in the position of the vertex associated with the largest dual variable cause the largest changes in the value of the SSH linear program.

2.8.2 Exact Intersection Point

Once the closest features have been determined from the dual solution to (2.29), this information can be used to predict the intersection point or to calculate it after contact has occurred through a simple projection operation. The exact operation depends on the type of (predicted) contact that has occurred.

2.9 Conclusion

This chapter has developed a novel approach to collision detection in the form of the supporting separating hyperplane (SSH) test for interpenetration, which can be formulated and solved as a linear program for polyhedral convex bodies as the SSH LP and offers advantages in friction formulations, collision integrators, and control algorithms. It is further shown that the subdifferential of the SSH LP exists, is local, and lies in the normal cone of a contact configuration. An explicit formulation for the subderivative is derived, which is general for all degenerate solutions of the linear program. Several examples illustrate the usefulness of the primal and dual solutions to (2.29) in determining closest features and projected collision points, and examples in Chapter 3 will shows its utility in determining the force system at the time of contact in dynamics simulations. In addition, Chapter 4 will show the usefulness of the SSH LP within the framework of discrete mechanics and optimal control [52], to the control of systems of multiple non-smooth or non-convex bodies that tend to–or need to–cluster, e.g., to execute docking or self-assembly maneuvers.

3 Dynamics with Non-Smooth Collisions

3.1 Introduction and Organization

The developments in Chapter 2 are motivated by the need for collision models that lead to robust simulations of rigid body systems involving many collisions between non-smooth bodies. After an overview of two relevant time-integration schemes (Newmark and Discrete Variational), this chapter develops and presents several options for the consideration of such systems, which can be roughly divided into three categories; potential-based (smooth) variational approaches, conservative discrete variational approaches, and conserving approximations, or decomposition-based approaches. This chapter will describe each of these approaches in turn, with an eye not only on modeling large self-assembling systems, but also as a step towards the work in Chapter 4, in which optimal control algorithms are developed.

In potential-based approaches, to be discussed in Section 3.3 the bodies in the system are allowed to overlap by a small amount, and a smooth approximation to the indicator function is used directly to introduce contact forces into the system. While this method benefits from straightforward implementation for any choice of integrator and fast run times, the time step is limited by the need to prevent the overlap of any two bodies from becoming too large–both to avoid non-physical results and to prevent numerical instability in the presence of a strong potential. For non-smooth bodies, this approach is directly enabled by the use of the SSH LP, with good physical results coming directly from the locality and normality of the SSH LP subgradient.

In conservative discrete variational approaches (c.f. Fetecau et al. [25], Gonzalez et al. [32]), *discrete* momenta and energy are exactly conserved through collision events by recourse to the exact determination of the each contact time and configuration, which lie between predesignated time nodes. Application of these methods to non-smooth bodies has been enabled by the development of the SSH LP, however for the particular systems of interest, in which may bodies clump together and multiple collisions must be resolved between time nodes, this type of method becomes prohibitively

expensive. These methods will be described in Section 3.4.

Perhaps most useful for optimal control formulations are the family of methods dubbed conserving approximations (c.f. Cirak and West [15], Kane et al. [38]), and extended in this work to multi rigid body systems. In these approaches, rather than exactly resolving the contact time and configuration, a contact time is assumed (e.g., at a time node node), and the contact configuration is determined by closest-point projection. Using this approximated configuration, energy and momentum are exactly conserved, in what is now an explicit calculation. In Section 3.5, the approach first developed by Cirak and West [15] for finite elements is extended to rigid body dynamics in in the context of a constrained discrete variational integrator developed by Leyendecker et al. [50].

3.2 Time Integration Overview

Throughout this chapter, two different time-integration schemes will be used to illustrate the potential, fully conservative, and decomposition-based approaches to modeling dynamics with collisions for Lagrangian systems of the form

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^T \boldsymbol{M}^{-1} \dot{\boldsymbol{q}} - \sum_k V_k(\boldsymbol{q}), \tag{3.1}$$

where V_k are a holonomic potentials, i.e. $V_k : \mathcal{Q} \to \mathbb{R}$, and $L : T\mathcal{Q} \to \mathbb{R}$. This section provides an overview of these approaches to time-integration.

Without loss of generality, a single potential $V(\boldsymbol{q})$ can be treated.

The first of these schemes, the Newmark method (c.f. [39]), is one of the most widely used schemes in forward dynamics, and in particular in finite element simulations. The second scheme introduced in this section is a constrained version of the discrete variational approach introduced (c.f. [11, 39, 50] and references therein), which is extremely useful for rigid body and multi-rigid body systems, and, in its forced version, for formulating and solving optimal control problems for these systems using the minimal set of variables. A general formulation for this scheme will be given in Section 3.2.2, and will be modified accordingly for the fully conservative and decomposition based integrators introduced and developed in subsequent sections.

3.2.1 Newmark-β

For state $\boldsymbol{q} \in \mathcal{Q}$, $\dot{\boldsymbol{q}} \in T_{\boldsymbol{q}}\mathcal{Q}$, time $t \in [0, T]$, and a Lagrangian of the form (3.1) with both $\mathcal{Q}$ and $T_{\boldsymbol{q}}\mathcal{Q} = \mathbb{R}^n$, enforcing stationarity of the action integral

$$I(\boldsymbol{q}) = \int_0^T L(\boldsymbol{q}, \dot{\boldsymbol{q}}) dt, \tag{3.2}$$

leads to the Euler-Lagrange equations

$$\boldsymbol{M}\ddot{\boldsymbol{q}} = -\nabla V(\boldsymbol{q}).$$

Letting $\boldsymbol{a} = -\boldsymbol{M}^{-1}\nabla V(\boldsymbol{q})$, with scalar constants $\gamma \in [0, 1]$, $\beta \in \left[0, \frac{1}{2}\right]$, and time step h, the Newmark family of schemes describes the evolution of the state $(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$ at times $t = kh$, $k = 0, \ldots, K$ as follows. Given state $(\boldsymbol{q}(kh), \dot{\boldsymbol{q}}(kh)) \approx (\boldsymbol{q}_k, \dot{\boldsymbol{q}}_k)$, the update $(\boldsymbol{q}_{k+1}, \dot{\boldsymbol{q}}_{k+1})$ is given by

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k + h\dot{\boldsymbol{q}}_k + \frac{h^2}{2}\left[(1 - 2\beta)\boldsymbol{a}_k + 2\beta\boldsymbol{a}_{k+1}\right] \tag{3.3a}$$

$$\dot{\boldsymbol{q}}_{k+1} = \dot{\boldsymbol{q}}_k + h\left[(1 - \gamma)\boldsymbol{a}_k + \gamma\boldsymbol{a}_{k+1}\right]. \tag{3.3b}$$

For $\beta = 0$, the system is explicit. The standard explicit scheme is given by $\gamma = \frac{1}{2}$ and $\beta = 0$, and the forward Euler method is given by $\gamma = 0$ and $\beta = 0$. If rotational dynamics are present, then the Newmark-based explicit method of Krysl and Endres [45], or any of Krysl [44]'s implicit methods can be used.

3.2.2 Discrete Variational Integrators

In contrast to the Newmark family of schemes, which discretized the continuous Euler-Lagrange equations, discrete variational integrators solve the *discrete* Euler-Lagrange equations, which are the result from enforcing stationarity for a discrete action sum. In particular, the action integral over one time period is approximated by $L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1})$ as

$$
\begin{aligned}
L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) &\approx \int_{t_k}^{t_{k+1}} L(\boldsymbol{q}, \dot{\boldsymbol{q}}) dt \\
L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) &= h \left[\frac{1}{2h^2} (\boldsymbol{q}_{k+1} - \boldsymbol{q}_k)^T \boldsymbol{M} (\boldsymbol{q}_{k+1} - \boldsymbol{q}_k) - V\left([1-\gamma]\, \boldsymbol{q}_{k+1} + \gamma \boldsymbol{q}_k \right) \right],
\end{aligned}
\tag{3.4}
$$

with $\gamma \in \left[0, \frac{1}{2}\right]$, so that the map $L_d : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$. Note that while (3.4) is used throughout this thesis with $\gamma = \frac{1}{2}$, is not a unique choice for L_d. Other approximations to the integral, such as the trapezoid rule, could also be used. The discrete action sum is then given by

$$
I_d = \sum_{k=0}^{K-1} L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}).
$$

Enforcing stationarity of the action sum with respect to the discrete trajectory $\boldsymbol{q}_d = \{\boldsymbol{q}_k\}_{k=0}^{K}$, with fixed end points yields the discrete Euler-Lagrange equations

$$
D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) + D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k) = \boldsymbol{0}.
$$

Using the midpoint rule (i.e., $\gamma = \frac{1}{2}$), the discrete Euler-Lagrange equations are given by

$$
\boldsymbol{M} \frac{1}{h^2} \left(\boldsymbol{q}_{k+1} - 2\boldsymbol{q}_k + \boldsymbol{q}_{k-1} \right) = -\nabla V \left(\frac{\boldsymbol{q}_{k+1} + \boldsymbol{q}_k}{2} \right) - \nabla V \left(\frac{\boldsymbol{q}_k + \boldsymbol{q}_{k-1}}{2} \right),
$$

and lead directly to a two-step method that can be solved for $\boldsymbol{q}_{k+1}$ given $(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k)$.

3.2.3 Constrained Discrete Variational Integrators

Extensive use will be made in this chapter of constrained mechanical systems, in which the $n-$dimensional trajectory is required to evolve on the submanifold $\mathcal{C} \subset \mathcal{Q}$ with $\mathcal{C}$ described by m holonomic constraints, $\boldsymbol{g}(\boldsymbol{q})$, i.e.,

$$
\mathcal{C} = \{ \boldsymbol{q} \in \mathcal{Q} | \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0} \} \subset \mathcal{Q}, \tag{3.5}
$$

with tangent bundle

$$T\mathcal{C} = \{(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in T\mathcal{Q} | \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}, \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}\} \subset T\mathcal{Q}. \tag{3.6}$$

With the introduction of constraints to the system, the Euler-Lagrange equations now extremize an augmented Lagrangian given by

$$\overline{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\lambda}) = L(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}^T(\boldsymbol{q})\boldsymbol{\lambda}.$$

Discretizing the additional terms involving the Lagrange multipliers and constraints according to

$$\frac{h}{2}\boldsymbol{g}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k + \frac{h}{2}\boldsymbol{g}^T(\boldsymbol{q}_{k+1})\boldsymbol{\lambda}_{k+1} \approx \int_{t_k}^{t_{k+1}} \boldsymbol{g}^T(\boldsymbol{q})\boldsymbol{\lambda} dt$$

gives rise to the augmented discrete Lagrangian

$$\overline{L}_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k, \boldsymbol{\lambda}_k, \boldsymbol{\lambda}_{k+1}) = L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) - \frac{h}{2}\boldsymbol{g}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k - \frac{h}{2}\boldsymbol{g}^T(\boldsymbol{q}_{k+1})\boldsymbol{\lambda}_{k+1} \tag{3.7}$$

in which a set of Lagrange multipliers, $\boldsymbol{\lambda}_d = \{\boldsymbol{\lambda}_k\}_{k=0}^K$, restrict the trajectory to the submanifold on which $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$. The corresponding action sum is given by

$$I_d = \sum_{k=0}^{K-1} \overline{L}_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{\lambda}_k, \boldsymbol{\lambda}_{k+1}).$$

Extremizing the action with respect to all variables–now including $\boldsymbol{\lambda}_d$–yields the *constrained* discrete Euler-Lagrange equations

$$\begin{aligned} D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) + D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k) - h\boldsymbol{G}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k &= \boldsymbol{0}, \\ \boldsymbol{g}(\boldsymbol{q}_{k+1}) &= \boldsymbol{0}, \end{aligned} \tag{3.8}$$

in which $\boldsymbol{G}$ is the Jacobian of the constraint vector $\boldsymbol{g}$. In this form, the constrained discrete Euler-Lagrange equations are an $(n+m)-$dimensional system to be solved for $\boldsymbol{q}_{k+2}$ and $\boldsymbol{\lambda}_{k+1}$, given $(\boldsymbol{q}_k, \boldsymbol{q}_{k+1})$. The term $-h\boldsymbol{G}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k$ represents the constraint forces which prevent the system from deviating from the constraint manifold, $\mathcal{C}$.

Discrete Null Space Approach. In the analog to the continuous null space approach (c.f. [9, 58]), Equations 3.8 can be reduced via premultiplication by a discrete null space matrix to eliminate the Lagrange multipliers (c.f. [50]). In the discrete case, the null space matrix $\boldsymbol{P}$ should satisfy

$$\text{range}(\boldsymbol{P}(\boldsymbol{q}_k)) = \text{null}(\boldsymbol{G}(\boldsymbol{q}_k)) = T_{\boldsymbol{q}_k}\mathcal{C}.$$

One such matrix, which will be called $\boldsymbol{Q}$, can be found from the natural orthogonal complement of $\boldsymbol{G}(\boldsymbol{q}_k)$. Another option is to use the projection given by

$$\boldsymbol{Q} = I_{n\times n} - \boldsymbol{G}^T \left[\boldsymbol{G}\boldsymbol{M}^{-1}\boldsymbol{G}^T\right]^{-1} \boldsymbol{G}\boldsymbol{M}^{-1}, \tag{3.9}$$

with $\boldsymbol{G}$ evaluated at the appropriate discrete configuration, $\boldsymbol{q}_k$. The null space matrix $\boldsymbol{Q}$ is a mapping $\boldsymbol{Q}(\boldsymbol{q}_k) : T^*_{\boldsymbol{q}_k}\mathcal{Q} \to \eta(T^*_{\boldsymbol{q}_k}\mathcal{C})$, in which is an embedding of the lower dimensional sub-manifold into the redundant cotangent bundle, i.e. $\eta : T^*\mathcal{C} \to T^*\mathcal{Q}$ (see [58] for more details). By premultiplication with $\boldsymbol{Q}(\boldsymbol{q}_k)$, Equation 3.8 becomes

$$\boldsymbol{Q}(\boldsymbol{q}_k)\left[D_1L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, h) + D_2L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k, h)\right] = \boldsymbol{0}, \tag{3.10a}$$

$$\boldsymbol{g}(\boldsymbol{q}_{k+1}) = \boldsymbol{0}. \tag{3.10b}$$

In 3.10, the Lagrange multipliers have been eliminated from the system, but the equations of motion remain a redundant $(n+m)-$dimensional system to be solved for n configuration variables.

A second null space matrix may be found directly from the definition of a null space matrix as the linear map from generalized velocities to admissible redundant velocities $\boldsymbol{P}(\boldsymbol{q}) : \mathbb{R}^{n-m} \to T_{\boldsymbol{q}}\mathcal{C}$. For example, the twist $(\boldsymbol{t})$ of a free rigid body may be described by its translational velocity, $\dot{\boldsymbol{\varphi}}$, and its angular velocity, $\boldsymbol{\omega}$, whereupon the redundant velocities, $\dot{\boldsymbol{q}}$, of the body may be expressed as

$$\begin{aligned} &\dot{\boldsymbol{q}} = \boldsymbol{P}\boldsymbol{\nu}, \text{ with:} \\ &\boldsymbol{\nu} = \begin{bmatrix} \dot{\boldsymbol{\varphi}} \\ \boldsymbol{\omega} \end{bmatrix}. \end{aligned} \tag{3.11}$$

The resulting integration scheme is given by

$$\boldsymbol{P}^T(\boldsymbol{q}_k)\left[D_1L_d(\boldsymbol{q}_k,\boldsymbol{q}_{k+1},h)+D_2L_d(\boldsymbol{q}_{k-1},\boldsymbol{q}_k,h)\right]=\boldsymbol{0}, \tag{3.12a}$$

$$\boldsymbol{g}(\boldsymbol{q}_{k+1})=\boldsymbol{0}. \tag{3.12b}$$

In fact, $\boldsymbol{P}^T(\boldsymbol{q})$ can be thought of as a map $\boldsymbol{P}^T(\boldsymbol{q}):T^*_{\boldsymbol{q}}\mathcal{Q}\to T^*_{\boldsymbol{u}}\mathcal{U}$, in which $\boldsymbol{u}\in\mathcal{U}$ are a minimal set of local coordinates in the generalized manifold $\mathcal{U}$, to be introduced in the following section.

Reparametrization in Local Coordinates. In addition to eliminating the Lagrange multipliers, the kinematic constraints $\boldsymbol{g}(\boldsymbol{q}_k)$ can be explicitly eliminated from (3.8) via a mapping from the minimal set of local coordinates $\boldsymbol{u}\in\mathbb{R}^m$ on the constraint manifold to the fully redundant coordinates $\boldsymbol{q}\in\mathbb{R}^n$, $\boldsymbol{F}_d:\mathcal{U}\mapsto\mathcal{Q}$, i.e., a local nodal reparametrization for the redundant configuration variables in terms of a minimal set of generalized variables. The local mapping $\boldsymbol{F}_d$, can be defined such that $\boldsymbol{q}_k=\boldsymbol{F}_d(\boldsymbol{u}_k,\boldsymbol{q}_{k-1})$ with $\boldsymbol{g}(\boldsymbol{F}_d(\boldsymbol{u}_k,\boldsymbol{q}_{k-1}))=\boldsymbol{0}$. It is readily shown that the Jacobian of $\boldsymbol{F}_d$ w.r.t. $\boldsymbol{u}$ satisfies

$$\text{range}\left(\frac{d\boldsymbol{F}_d(\boldsymbol{u}_{k+1},\boldsymbol{q}_k)}{d\boldsymbol{u}}\right)=\text{null}(\boldsymbol{G}(\boldsymbol{q}_k)),$$

however it is most convenient to use this reparametrization in conjunction with the null space matrix introduced in Equation (3.11) to arrive at an integration scheme which eliminates the need to solve for the Lagrange multipliers and explicitly enforces the constraints as

$$\boldsymbol{P}^T(\boldsymbol{q}_k)\left[D_1L_d(\boldsymbol{q}_k,\boldsymbol{F}_d(\boldsymbol{u}_{k+1},\boldsymbol{q}_k),h)+D_2L_d(\boldsymbol{q}_{k-1},\boldsymbol{q}_k,h)\right]=\boldsymbol{0}. \tag{3.13}$$

Constrained Discrete Legendre Transform. An alternative time-stepping scheme may be formulated using discrete momenta obtained from the constrained discrete Legendre transforms, $\mathbb{F}^{c^-}L_d{:}\mathcal{Q}\times\mathcal{Q}\to T^*_{\boldsymbol{q}_k}Q$ and $\mathbb{F}^{c^+}L_d{:}\mathcal{Q}\times\mathcal{Q}\to T^*_{\boldsymbol{q}_k}Q$, which are defined as

$$\begin{aligned}
&\mathbb{F}^{c^-}:(\boldsymbol{q}_k,\boldsymbol{q}_{k+1})\mapsto(\boldsymbol{q}_k,\boldsymbol{p}_k^-)\\
&\boldsymbol{p}_k^-=-D_1L_d(\boldsymbol{q}_k,\boldsymbol{q}_{k+1},h)+\frac{h}{2}\boldsymbol{G}^T(\boldsymbol{q}_k)\cdot\boldsymbol{\lambda}_k\\
&\mathbb{F}^{c^+}:(\boldsymbol{q}_{k-1},\boldsymbol{q}_k)\mapsto(\boldsymbol{q}_k,\boldsymbol{p}_k^+)\\
&\boldsymbol{p}_k^+=D_2L_d(\boldsymbol{q}_{k-1},\boldsymbol{q}_k,h)-\frac{h}{2}\boldsymbol{G}^T(\boldsymbol{q}_k)\cdot\boldsymbol{\lambda}_k.
\end{aligned} \tag{3.14}$$

The resulting scheme to advance the state of the system from $(\boldsymbol{q}_k, \boldsymbol{p}_k)$ to $(\boldsymbol{q}_{k+1}, \boldsymbol{p}_{k+1})$ (along with $\boldsymbol{\lambda}_{k-1}$ and $\boldsymbol{\lambda}_k$, respectively) is then

$$\begin{aligned} \boldsymbol{p}_k^+ - \boldsymbol{p}_k^- &= \mathbf{0} \\ \boldsymbol{g}(\boldsymbol{q}_{k+1}) &= \mathbf{0}, \end{aligned} \tag{3.15}$$

with $\boldsymbol{p}_{k+1}$ then updated according to (3.14). The integrator resulting from the discrete Legendre transformation will be referred to as the $(\boldsymbol{q}, \boldsymbol{p})$ scheme, as opposed to the original $(\boldsymbol{q}, \boldsymbol{q})$ scheme.

With the definition of the discrete Legendre transform in hand, it is clear that Equations (3.8) and (3.12) may be interpreted as enforcing the balance of discrete momenta at each time node, $\boldsymbol{p}_k^+ - \boldsymbol{p}_k^- = \mathbf{0}$. Likewise, two additional mappings which eliminate the Lagrange multipliers from (3.14), ${}^{Q}\mathbb{F}^c L_d{:}\mathcal{Q} \times \mathcal{Q} \to \eta(T^*_{\boldsymbol{q}_k}\mathcal{C})$ and ${}^{P}\mathbb{F}^c L_d{:}\mathcal{Q} \times \mathcal{Q} \to T^*\mathcal{U}$ may be defined as follows

$$\begin{aligned} {}^{Q}\boldsymbol{p}_k^- &= \boldsymbol{Q}(\boldsymbol{q}_k)\left[-D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1})\right] \\ {}^{Q}\boldsymbol{p}_k^+ &= \boldsymbol{Q}(\boldsymbol{q}_k)\left[D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k)\right], \end{aligned} \tag{3.16}$$

and

$$\begin{aligned} {}^{P}\boldsymbol{p}_k^- &= \boldsymbol{P}^T(\boldsymbol{q}_k)\left[-D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1})\right] \\ {}^{P}\boldsymbol{p}_k^+ &= \boldsymbol{P}^T(\boldsymbol{q}_k)\left[D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k)\right], \end{aligned} \tag{3.17}$$

which will be referred to as the projected scheme (3.16) and the reduced scheme (3.17), respectively.

Thus, all that is required to start the $(\boldsymbol{q}, \boldsymbol{p})$ time-stepping scheme is the pair, $(\boldsymbol{q}_0, \boldsymbol{\nu}_0)$ from which $(\boldsymbol{q}_0, \boldsymbol{p}_0)$ may be calculated according to (3.11) and by letting $\boldsymbol{p}_0 = \boldsymbol{M}\dot{\boldsymbol{q}}_0$. For example, using Equation (3.17), $\boldsymbol{q}_1$ may be solved from ${}^{P}\boldsymbol{p}_0^- = P^T\boldsymbol{p}_0$, ${}^{P}\boldsymbol{p}_1^+$ set from $(\boldsymbol{q}_0, \boldsymbol{q}_1)$, and so on. It is important to note that the Lagrange multipliers are not an immediate output of a $(\boldsymbol{q}, \boldsymbol{p})$ scheme which uses either the projected or reduced discrete Legendre transforms (3.12,3.17), however the term involving the gradient of the constraint forces *is* a part of the total, unprojected momentum. Furthermore, when using either the reduced or constrained schemes, the Lagrange multipliers, $\boldsymbol{\lambda}_n$,

are not available until $\boldsymbol{q}_{k+1}$ or $\boldsymbol{p}_{k+1}$ have been obtained. At this point, we may recover $\boldsymbol{\lambda}_k$ via

$$\begin{aligned} \boldsymbol{\lambda}_k &= \boldsymbol{R}_d^T(\boldsymbol{q}_k)\left[D_2 L_d(\boldsymbol{q}_{k-1},\boldsymbol{q}_k) + D_1 L_d(\boldsymbol{q}_k,\boldsymbol{q}_{k+1})\right] \\ \boldsymbol{R}_d &= \frac{1}{h}\boldsymbol{G}^T(\boldsymbol{q}_k)\cdot\left[\boldsymbol{G}(\boldsymbol{q}_k)\cdot\boldsymbol{G}^T(\boldsymbol{q}_k)\right]^{-1}. \end{aligned} \tag{3.18}$$

Hidden Constraints. A full explanation of this topic may be found in [50]. The aforementioned time-stepping schemes do not enforce a set of hidden constraints on the redundant momenta which come from the temporary differentiated form of the constraint vector $\boldsymbol{g}$. On the momentum level, these constraints constraints read

$$\boldsymbol{G}(\boldsymbol{q}_k)\boldsymbol{M}^{-1}\boldsymbol{p}_k = \boldsymbol{0}. \tag{3.19}$$

As can be seen in [50], the discrete momenta, including the Lagrange multiplier terms, resulting from (3.15) solved with the projected or reduced momenta (3.17,3.16) do not precisely fulfill these constraints in the presence of a potential. That is, they do not lie in the null space of the internal constraint Jacobian. However, the projected momenta ${}^{Q}\boldsymbol{p}_k$, and the redundant moment recovered from the reduced scheme,

$$\begin{aligned} \boldsymbol{p}_k &= \boldsymbol{M}\boldsymbol{P}(\boldsymbol{q}_k)\boldsymbol{M}_{red}^{-1})(\boldsymbol{q}_k)\ {}^{P}\boldsymbol{p}_k, \text{ with} \\ \boldsymbol{M}_{red}(\boldsymbol{q}_k) &= \boldsymbol{P}^T(\boldsymbol{q}_k)\boldsymbol{M}\boldsymbol{P}(\boldsymbol{q}_k), \end{aligned} \tag{3.20}$$

fulfill the discrete hidden momentum constraints exactly.

3.2.4 Constrained Rigid Multi-Body Kinematics

The specification of the *constrained* equations of motion, and the corresponding discrete variational integrator, leads to a straightforward way of describing both free rigid bodies and kinematic chains. The examples and systems of interest in this chapter, and in Chapter 4 are largely concerned with either free rigid bodies, or systems of rigid bodies connected by joints. This section begins by precisely defining m_{int} rigid body constraints, $\boldsymbol{g}_{int}$, and the corresponding null space matrix, $\boldsymbol{P}_{int}$, which will be relevant to every body in the systems of interest. For kinematic chains, an additional set of m_{ext}^C constraints, $\boldsymbol{g}_{ext}^C$, are present and depend on the specification of the joint type(s) (e. g.

spherical, revolute, cylindrical, etc., see Betsch and Leyendecker [10] or Leyendecker [48] for a full enumeration of low order kinematic pairs), so that the total number of constraints is $m = m_{int} + m_{ext}^{C}$. The complete set of constraints is then given by

$$\boldsymbol{g} = \begin{bmatrix} \boldsymbol{g}_{int}(\boldsymbol{q}) \\ \boldsymbol{g}_{ext}^{C}(\boldsymbol{q}) \end{bmatrix} \in \mathbb{R}^{m},$$

with the constraint Jacobian analogously modified.

The determination of $\boldsymbol{g}_{ext}^{C} = \boldsymbol{g}_{ext}^{J}$ due to a joint of type J for simple kinematic pairs is relatively straightforward, but additional care must be given to the determination of the null space matrix for systems with joints. In particular, the reduction of the degrees number generalized coordinates from $n - m_{int}$ to $n - (m_{int} + m_{ext}^{J})$ leads to a corresponding reduction in the generalized velocities, which are now given by

$$\boldsymbol{\nu}^{J} = \begin{bmatrix} \boldsymbol{t}^{1} \\ \boldsymbol{\tau}^{J} \end{bmatrix} \in \mathbb{R}^{m},$$

where $\boldsymbol{\tau}^{J}$ are the $6 - m_{ext}^{J}$ joint velocities of the second body in the chain, and $\boldsymbol{t}^{1}$ is the full rigid body twist of the first body. The joint velocities are defined so that the twist of the second body can be written as

$$\boldsymbol{t}^{2,J} = \boldsymbol{P}_{ext}^{2,J}(\boldsymbol{q}) \cdot \boldsymbol{\nu}^{J}, \quad \text{with:} \quad \boldsymbol{P}_{ext}^{2,J}(\boldsymbol{q}) \in \mathbb{R}^{6 \times (12 - m_{ext}^{J})},$$

with the corresponding mapping for all redundant twist components given by

$$\boldsymbol{t}^{J} = \begin{bmatrix} \boldsymbol{t}^{1} \\ \boldsymbol{t}^{2,J} \end{bmatrix} = \boldsymbol{P}_{ext}^{J}(\boldsymbol{q}) \cdot \boldsymbol{\nu}^{J} = \begin{bmatrix} \boldsymbol{I}_{6\times 6} & \boldsymbol{0}_{6\times(6-m_{ext})} \\ & \boldsymbol{P}_{ext}^{2,J}(\boldsymbol{q}) \end{bmatrix} \boldsymbol{\nu}^{J}.$$

By composition of $\boldsymbol{P}_{ext}^{J}$ and $\boldsymbol{P}_{int}$, the null space matrix for the full system satisfying $\boldsymbol{P}^{T}(\boldsymbol{q}) : T_{\boldsymbol{q}}^{*}\mathcal{Q} \to T_{\boldsymbol{u}}^{*}\mathcal{U}$ is given by

$$\boldsymbol{P}^{J}(\boldsymbol{q}) = \boldsymbol{P}_{int}(\boldsymbol{q}) \cdot \boldsymbol{P}_{ext}^{J}(\boldsymbol{q}).$$

Rigid Body Constraints. For rigid body dynamics in three dimensions, a director formulation is used (see Figure 3.1) so that for each body $\boldsymbol{q}^T = \left[\boldsymbol{\varphi}^T \, \boldsymbol{d}_1^T \, \boldsymbol{d}_2^T \, \boldsymbol{d}_3^T\right] \in \mathbb{R}^{12}$. By the use of this formulation, singularities associates with other parametrizations (i.e. euler angles, quaternions, etc.) are completely avoided. Furthermore, the utility of the constrained integrator comes from the fact that all configuration variables *are in* $\mathbb{R}^n$, which simplifies the description of joint locations and the derivation of gradients of potentials, as well as explicit formulations for Jacobian and Hessian matrices that may be used to solve for the update.

The directors, $\boldsymbol{d}_I \in \mathbb{R}^3$ correspond to the columns of a rotation matrix in $SO(3)$ and therefore must satisfy the orthonormality constraints

$$\boldsymbol{g}_{int}(\boldsymbol{q}) = \begin{bmatrix} \frac{1}{2}(\boldsymbol{d}_1^T \boldsymbol{d}_1 - 1) \\ \frac{1}{2}(\boldsymbol{d}_2^T \boldsymbol{d}_2 - 1) \\ \frac{1}{2}(\boldsymbol{d}_3^T \boldsymbol{d}_3 - 1) \\ \boldsymbol{d}_1^T \boldsymbol{d}_2 \\ \boldsymbol{d}_1^T \boldsymbol{d}_3 \\ \boldsymbol{d}_2^T \boldsymbol{d}_3 \end{bmatrix} \in \mathbb{R}^6, \tag{3.21}$$

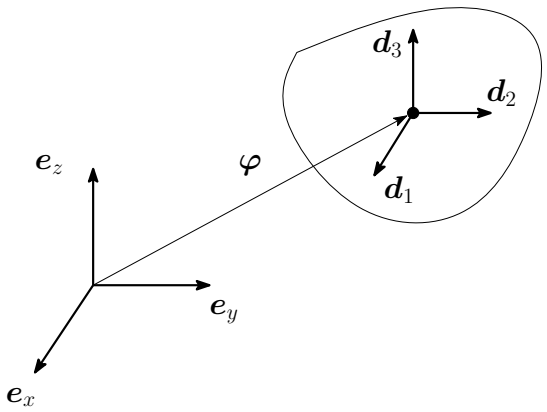


Figure 3.1: Rigid body coordinate system with redundant configuration variables (from Betsch and Leyendecker [10]).

and $\boldsymbol{\varphi} \in \mathbb{R}^3$ describes the translation of the center of mass of the body. The Jacobian of the rigid body constraints $\boldsymbol{g}_{int}$, $\boldsymbol{G}_{int}$, is given by

$$\boldsymbol{G}_{int}(\boldsymbol{q}) = \begin{bmatrix} & \begin{bmatrix} \boldsymbol{d}_1^T \\ \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \end{bmatrix} & \begin{bmatrix} \mathbf{0}_{3\times1} \\ \boldsymbol{d}_2^T \\ \mathbf{0_{3\times1}} \end{bmatrix} & \begin{bmatrix} \mathbf{0}_{3\times1} \\ \mathbf{0}_{3\times1} \\ \boldsymbol{d}_3^T \end{bmatrix} \\ \mathbf{0}_{3\times3} & & & \\ & \begin{bmatrix} \boldsymbol{d}_2^T \\ \boldsymbol{d}_3^T \\ \mathbf{0}_{3\times1} \end{bmatrix} & \begin{bmatrix} \boldsymbol{d}_1^T \\ \mathbf{0}_{3\times1} \\ \boldsymbol{d}_3^T \end{bmatrix} & \begin{bmatrix} \mathbf{0}_{3\times1} \\ \boldsymbol{d}_1^T \\ \boldsymbol{d}_2^T \end{bmatrix} \\ \mathbf{0}_{3\times3} & & & \end{bmatrix} \in \mathbb{R}^{6\times12}. \tag{3.22}$$

For this set of redundant variables, the appropriate (diagonal) mass matrix, $\boldsymbol{M}$, is given by

$$\boldsymbol{M} = \begin{bmatrix} m_{\boldsymbol{\varphi}}\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & E_1\boldsymbol{I} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & E_2\boldsymbol{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & E_3\boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{12\times12}, \tag{3.23}$$

in which m_φ denotes the mass of the body, and (E_1, E_2, E_3) are the principal values of the body's Euler tensor, $\boldsymbol{E}$, which can be found from the principal inertial tensor, $\boldsymbol{J}_0$, as $\boldsymbol{E} = \frac{1}{2}\operatorname{tr}(\boldsymbol{J}_0) - \boldsymbol{J}_0$.

The generalized coordinates are given by $\boldsymbol{u} \in \mathbb{R}^6$, with

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_\varphi \\ \boldsymbol{u}_T \end{bmatrix},$$

and the local reparametrization is given by

$$\boldsymbol{F}_d(\boldsymbol{u}_k, \boldsymbol{q}_{k-1}) = \begin{bmatrix} \boldsymbol{\varphi}_{k-1} + \boldsymbol{u}_{\varphi,k} \\ \exp(\widehat{\boldsymbol{u}_{T,k}})\boldsymbol{d}_{1,k-1} \\ \exp(\widehat{\boldsymbol{u}_{T,k}})\boldsymbol{d}_{2,k-1} \\ \exp(\widehat{\boldsymbol{u}_{T,k}})\boldsymbol{d}_{3,k-1} \end{bmatrix}. \tag{3.24}$$

Equation (3.24) makes use of the exponential mapping $\exp : \mathfrak{so}(3) \to SO(3)$, and the hat map, $\widehat{\cdot}$ is defined so that $\widehat{\boldsymbol{w}}\boldsymbol{d} = \boldsymbol{w} \times \boldsymbol{d}$. To recapitulate the example in the previous section, the redundant velocities $\dot{\boldsymbol{q}}$ can be written in terms of the generalized velocities $\boldsymbol{\nu}$–which for otherwise unconstrained systems correspond to the twist $\boldsymbol{t}$ for each body–as

$$\dot{\boldsymbol{q}} = \boldsymbol{P}\boldsymbol{\nu}, \text{ with:}$$
$$\boldsymbol{\nu} = \begin{bmatrix} \dot{\boldsymbol{\varphi}} \\ \boldsymbol{\omega} \end{bmatrix} \in \mathbb{R}^6,$$

with $\boldsymbol{P}_{int}(\boldsymbol{q})$ given by

$$\boldsymbol{P}_{int}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \widehat{-\boldsymbol{d}_1} \\ \boldsymbol{0} & \widehat{-\boldsymbol{d}_2} \\ \boldsymbol{0} & \widehat{-\boldsymbol{d}_3} \end{bmatrix} \in \mathbb{R}^{12\times 6}.$$

Spherical Pair. When two bodies with configurations $\boldsymbol{q}$ as described in the previous paragraph are linked by a spherical joint (see Figure 3.2), all relative translation between the bodies is removed, but both bodies remain free to rotate. This introduces an additional set of $m^S_{ext} = 3$ constraints given by

$$\boldsymbol{g}^S_{ext}(\boldsymbol{q}) = \boldsymbol{\varphi}^2 - \boldsymbol{\varphi}^1 + \boldsymbol{\varrho}^2 - \boldsymbol{\varrho}^1 = \boldsymbol{0} \in \mathbb{R}^3. \tag{3.25}$$

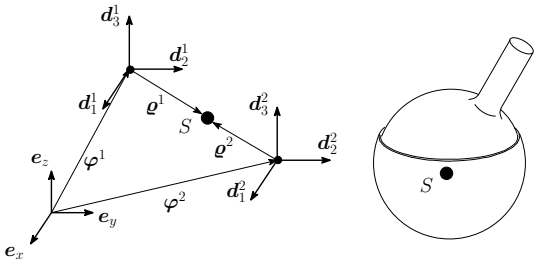


Figure 3.2: Two Rigid bodies linked by a spherical joint.

in (3.25), the vectors $\boldsymbol{\varrho}^j$ are given in terms of the directors, i.e. $\boldsymbol{\varrho} = \varrho_I \boldsymbol{d}_I$ in which repeated indices indicate the sum $\boldsymbol{\varrho} = \varrho_1\boldsymbol{d}_1 + \varrho_2\boldsymbol{d}_2 + \varrho_3\boldsymbol{d}_3$. For the pair shown in Figure 3.2, the generalized degrees of freedom are

$$\boldsymbol{u}^S = \begin{bmatrix} \boldsymbol{u}^1_\varphi \\ \boldsymbol{u}^1_T \\ \boldsymbol{u}^2_T \end{bmatrix} \in \mathbb{R}^9.$$

From the constraints in (3.25), the redundant configuration variables for the second body are given by the mapping

$$\boldsymbol{q}_k^2 = \boldsymbol{F}_d^{S,2}(\boldsymbol{u}_k^S, \boldsymbol{q}_{k-1}) = \begin{bmatrix} \boldsymbol{\varphi}_{k-1}^1 + \boldsymbol{u}_{\varphi,k}^1 + \exp(\widehat{\boldsymbol{u}_{T,k}^1})\boldsymbol{\varrho}_{k-1}^1 - \exp(\widehat{\boldsymbol{u}_{T,k}^2})\boldsymbol{\varrho}_{k-1}^2 \\ \exp(\widehat{\boldsymbol{u}_{T,k}^2})\boldsymbol{d}_{1,k-1}^2 \\ \exp(\widehat{\boldsymbol{u}_{T,k}^2})\boldsymbol{d}_{2,k-1}^2 \\ \exp(\widehat{\boldsymbol{u}_{T,k}^2})\boldsymbol{d}_{3,k-1}^2 \end{bmatrix}. \tag{3.26}$$

To determine a null space matrix for the spherical pair, note that the restriction on the configuration variables leads to a constraint on the relationship between the generalized velocities of the bodies,

$$\boldsymbol{\nu}^S = \begin{bmatrix} \dot{\boldsymbol{\varphi}}^1 \\ \boldsymbol{\omega}^1 \\ \boldsymbol{\omega}^2 \end{bmatrix} = \begin{bmatrix} \boldsymbol{t}^1 \\ \boldsymbol{\omega}^2 \end{bmatrix} \in \mathbb{R}^9.$$

Specifically,

$$\dot{\boldsymbol{\varphi}}^2 = \dot{\boldsymbol{\varphi}}^1 + \boldsymbol{\omega}^1 \times \boldsymbol{\varrho}^1 - \boldsymbol{\omega}^2 \times \boldsymbol{\varrho}^2. \tag{3.27}$$

Recall that the null space matrix must satisfy $\boldsymbol{t}^{2,S} = \boldsymbol{P}^{2,S}(\boldsymbol{q}) \cdot \boldsymbol{\nu}^S$, from which it can be deduced that

$$\boldsymbol{P}_{ext}^{2,S}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{I}_{3\times3} & -\widehat{\boldsymbol{\varrho}^1} & \widehat{\boldsymbol{\varrho}^2} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} \in \mathbb{R}^{6\times9}, \tag{3.28}$$

with the complete null space matrix for the system is $\boldsymbol{P}^S(\boldsymbol{q})$

$$\boldsymbol{P}^S(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{P}_{int}^1(\boldsymbol{q})\ \boldsymbol{0}_{12\times3} \\ \boldsymbol{P}_{int}^{2,S}(\boldsymbol{q}) \cdot \boldsymbol{P}_{ext}^{2,S}(\boldsymbol{q}) \end{bmatrix} \in \mathbb{R}^{24\times9}.$$

Revolute Pair. In a revolute pair, in addition to the relative translation between the bodies being restricted, the relative rotation is also restricted so that the bodies may only rotate relative to each other about a prescribed axis, as prescribed by the $m_{ext}^R = 5$ relationships in Equation 3.29.

This is illustrated in Figure 3.3 with the joint axis specified in terms of the first body's directors, $\boldsymbol{n}^1 = n_I^1 \boldsymbol{d}_I^1$.

$$\boldsymbol{g}_{ext}^R(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{\varphi}^2 - \boldsymbol{\varphi}^1 + \boldsymbol{\varrho}^2 - \boldsymbol{\varrho}^1 \\ (\boldsymbol{n}^1)^T \boldsymbol{d}_1^2 - \eta_1 \\ (\boldsymbol{n}^1)^T \boldsymbol{d}_2^2 - \eta_2 \end{bmatrix} = \boldsymbol{0} \in \mathbb{R}^5. \tag{3.29}$$

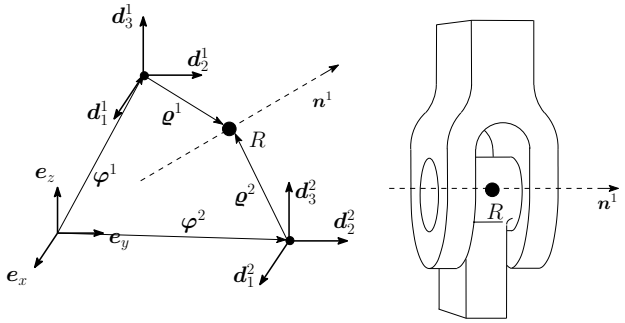


Figure 3.3: Two Rigid bodies linked by a revolute joint with axis $\boldsymbol{n}^1$.

In (3.29), $\eta_1, \eta_2 \in \mathbb{R}$ are constants determined by the initial geometry of the problem, and the final two constraints are not a unique choice and thus may be changed to avoid the occurrence of a singular null space matrix if $\boldsymbol{n}^1$ is collinear with either director or linearly dependent on the directors used in the formulation of the constraint. The generalized degrees of freedom are then

$$\boldsymbol{u}^R = \begin{bmatrix} \boldsymbol{u}_\varphi^1 \\ \boldsymbol{u}_T^1 \\ \theta^2 \end{bmatrix} \in \mathbb{R}^7,$$

in which θ^2 describes the incremental rotation of the second body about the axis $\boldsymbol{n}^1$. This formulation leads to the incremental update

$$\begin{aligned} \boldsymbol{q}_k^2 &= \boldsymbol{F}_d^{R,2}(\boldsymbol{u}_k^R, \boldsymbol{q}_{k-1}) \\ &= \begin{bmatrix} \boldsymbol{\varphi}_{k-1}^1 + \boldsymbol{I}_{k-1}^{11} \boldsymbol{u}_{\varphi,k}^1 + \exp(\widehat{\boldsymbol{u}_{T,k}^1}) \left[\boldsymbol{\varrho}_{k-1}^1 - \exp(-\widehat{N_{k-1}^{11} \boldsymbol{u}_{d,k}^1}) \exp(\widehat{\theta_k^2 \boldsymbol{n}_{k-1}^1}) \boldsymbol{\varrho}_{k-1}^2 \right] \\ \exp(-\widehat{N_{k-1}^{11} \boldsymbol{u}_{T,k}^1}) \exp(\widehat{\theta_k^2 \boldsymbol{n}_{k-1}^1}) \boldsymbol{d}_{1,k-1}^2 \\ \exp(-\widehat{N_{k-1}^{11} \boldsymbol{u}_{T,k}^1}) \exp(\widehat{\theta_k^2 \boldsymbol{n}_{k-1}^1}) \boldsymbol{d}_{2,k-1}^2 \\ \exp(-\widehat{N_{k-1}^{11} \boldsymbol{u}_{T,k}^1}) \exp(\widehat{\theta_k^2 \boldsymbol{n}_{k-1}^1}) \boldsymbol{d}_{3,k-1}^2 \end{bmatrix}, \end{aligned} \tag{3.30}$$

In which $\boldsymbol{I}_{k-1}^{11} = \boldsymbol{I} - N_{k-1}^{11}$ and $N_{k-1}^{11} = \boldsymbol{n}_{k-1}^1 \otimes \boldsymbol{n}_{k-1}^1$. In terms of the generalized velocities

$$\boldsymbol{\nu}^R = \begin{bmatrix} \boldsymbol{t}^1 \\ \dot{\theta}^2 \end{bmatrix} \in \mathbb{R}^7,$$

the angular velocity of the second body must be related to the angular velocity of the first body by

$$\boldsymbol{\omega}^2 = (\boldsymbol{I} - N^{11}) \cdot \boldsymbol{\omega}^1 + \dot{\theta^2}\boldsymbol{n}^1, \tag{3.31}$$

so that the translational velocity of the second body is determined by the relationship

$$\dot{\boldsymbol{\varphi}}^2 = \dot{\boldsymbol{\varphi}}^1 + \boldsymbol{\omega}^1 \times (\boldsymbol{\varrho}^1 - \boldsymbol{\varrho}^2) + \left[N^{11} \cdot \boldsymbol{\omega}^1\right] \times \boldsymbol{\varrho}^2 + \dot{\theta}^2 \boldsymbol{\varrho}^2 \times \boldsymbol{n}^1. \tag{3.32}$$

These relationships along with $\boldsymbol{t}^{2,R} = \boldsymbol{P}_{ext}^{2,R}(\boldsymbol{q}) \cdot \boldsymbol{\nu}^R$ can be used to derive

$$\boldsymbol{P}_{ext}^{2,R}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{I}_{3\times3} & \widehat{\boldsymbol{\varrho}^2 - \boldsymbol{\varrho}^1} - \widehat{\boldsymbol{\varrho}^2} \cdot N^{11} & \boldsymbol{\varrho}^2 \times \boldsymbol{n}^1 \\ \boldsymbol{0}_{3\times3} & \boldsymbol{I} - \boldsymbol{n}^1 \otimes \boldsymbol{n}^1 & \boldsymbol{n}^1 \end{bmatrix} \in \mathbb{R}^{6\times7}, \tag{3.33}$$

with the complete null space matrix for the pair given by

$$\boldsymbol{P}^R(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{P}_{int}^1(\boldsymbol{q}) & \boldsymbol{0}_{12\times1} \\ \boldsymbol{P}_{int}^2(\boldsymbol{q}) \cdot \boldsymbol{P}_{ext}^{2,R}(\boldsymbol{q}) \end{bmatrix}.$$

3.3 Potential-Based Contact Dynamics

In potential-based approaches, the dynamics of colliding bodies are modeled by the introduction of a parameter-dependent contact potential, $V_{\mathcal{A}}$, which constitutes a smooth approximation to the indicator function of the admissible (non-interpenetrating) set of configurations.

3.3.1 Non-Smooth Equations of Motion

In the non-smooth setting, contact is accounted for via the introduction of an indicator function of the set of admissible configurations directly into the Lagrangian of the system. Here, $\mathcal{Q}$ is a

configuration manifold and $T_{\boldsymbol{q}}\mathcal{Q}$ is the tangent space to $\mathcal{Q}$ at $\boldsymbol{q}$, i.e. the state variables consist of configurations $\boldsymbol{q} \in \mathcal{Q}$ and velocities $\dot{\boldsymbol{q}} \in T_{\boldsymbol{q}}\mathcal{Q}$ (see Marsden and Ratiu [57]). Admissible (non-contact) configurations for $\boldsymbol{q}$ occupy the subset $\mathcal{A} \subset \mathcal{Q}$. Neglecting, for simplicity, any potentials present in the system in the system, the Lagrangian becomes

$$L(\boldsymbol{q}, \dot{\boldsymbol{q}}) = \dot{\boldsymbol{q}}^T \boldsymbol{M} \dot{\boldsymbol{q}} - I_{\mathcal{A}}(\boldsymbol{q}), \text{ with:} \tag{3.34a}$$

$$I_{\mathcal{A}}(\boldsymbol{q}) = \begin{cases} 0 & \text{if } \boldsymbol{q} \in \mathcal{A} \subset \mathcal{Q} \\ \infty & \text{otherwise.} \end{cases} \tag{3.34b}$$

Enforcing stationarity of the action integral (3.2) leads to the equations of motion

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \partial I_{\mathcal{A}}(\boldsymbol{q}) \ni \boldsymbol{0}. \tag{3.35}$$

In (3.35), $\partial I_{\mathcal{A}}(\boldsymbol{q})$ denotes the *generalized differential* of the indicator function (c.f. [16, 38]), which is defined in Section 2.6. It is readily shown (c.f. [16]) that the generalized differential of the indicator function of a set is given by the normal cone, $N_{\mathcal{A}}$, of the set

$$\partial I_{\mathcal{A}}(\boldsymbol{q}) = N_{\mathcal{A}}(\boldsymbol{q}). \tag{3.36}$$

Comment on Implicit-Explicit Schemes. In [38], an implicit-explicit variational collision integrator is formulated in which inadmissible predicted configurations, $\boldsymbol{q}_{pre}$ are updated to admissible configurations $\boldsymbol{q}$, and then contact forces are applied to the system according to the potential

$$V_{\mathcal{A}}(\boldsymbol{q}, \boldsymbol{q}_{pre}) = \frac{1}{h^2}(\boldsymbol{q} - \boldsymbol{q}_{pre})^T \boldsymbol{M}(\boldsymbol{q} - \boldsymbol{q}_{pre}). \tag{3.37}$$

This method still essentially depends on the parameter $\frac{1}{h^2}$, and is thus also prone to instability issues if the amount of overlap at $\boldsymbol{q}_{pre}$ is too great for a given choice of step size.

3.3.2 Contact Potential

A parameter dependent contact potential based on a smooth approximation to the indicator function for each pairwise interaction can be formulated as

$$I_{\mathcal{A},i}(\boldsymbol{q}) \approx V_{\mathcal{A},i}(\boldsymbol{q}) = \begin{cases} 0 & \text{if } g_i(\boldsymbol{q}) < 0 \\ \frac{C}{2} g_i(\boldsymbol{q})^2 & \text{otherwise} \end{cases}, \tag{3.38}$$

the i^{th} pair of convex bodies in the system, with $C \in \mathbb{R}$ as a parameter to be chosen based on the time step, other potentials in the system, and the required accuracy (i.e., maximum allowable overlap). Collision integrators based simply on the introduction of $V_{\mathcal{A}}(\boldsymbol{q})$ can be quite effective, in particular if $\partial g(\boldsymbol{q}) \approx N_{\mathcal{A}}(\boldsymbol{q})$. However, the dependence on the parameter C limits the accuracy and applicability of the method, in large part due to the limitations on time step, h, that a large parameter C places on discrete time-integration algorithms. Let I_c index all possible pairwise contact interactions in a system. Then the complete contact potential for the full system is given by

$$V_{\mathcal{A}}(\boldsymbol{q}) = \sum_{i \in I_c} V_{\mathcal{A},i}(\boldsymbol{q}). \tag{3.39}$$

Contact Potential Subgradient. For lumped mass finite element models in which the degrees of freedom consist of the nodal coordinates, the definition of the SSH LP subgradient given in Section 2.7 is sufficient for implementation. For rigid body models, the chain rule must be used to recover $\partial V_{\mathcal{A}}(\boldsymbol{q})$.

Recall Equation (2.29) from Chapter 2, reproduced below

$$\begin{aligned} g(K_1, K_2) = - \max_{\substack{\boldsymbol{\alpha} \in \mathbb{R}^n \\ a_1, a_2 \in \mathbb{R}}} \; & a_1 - a_2, \text{ Subject to:} \\ & \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle - a_1 \geq 0, \;\; \boldsymbol{x} \in \operatorname{ext} K_1 \\ & \langle \boldsymbol{\alpha}, \boldsymbol{x} \rangle - a_2 \leq 0, \;\; \boldsymbol{x} \in \operatorname{ext} K_2 \\ & \left\langle \hat{\boldsymbol{\beta}}, \boldsymbol{\alpha} \right\rangle - 1 = 0. \end{aligned}$$

In (2.29), ext denotes the extreme points of the set, which in this case constitute the vertices of the set. The function $g(K_1, K_2)$ will be positive (i.e., the max function will be negative) if the sets are interpenetrating, and negative (the max function will be positive) if the are not interpenetrating. This concept is illustrated in Figure 2.7. This function has been reduced from a quadratically constrained linear program by changing the final constraint from $\boldsymbol{\alpha} \in S^{n-1}$ to the linear constraint shown by the selection of vector, $\hat{\boldsymbol{\beta}}$, which should be selected to maintain the invariance of the SSH QP, with more details to be given shortly.

As shown in Chapter 2, the generalized derivative of the SSH LP satisfies $\partial g \in N_{\mathcal{A}}$ for configurations in which $g(\boldsymbol{q}) = 0$. Furthermore, if the set of optimal primal solutions to (2.29) for the variables $\boldsymbol{\alpha}$ is denoted $\boldsymbol{A}_{\alpha}^{*}$, and the set of optimal dual solutions to (2.29) is denoted $\boldsymbol{\Pi}^{*}$, then a direction $\nabla_{\boldsymbol{x}_i} g \subset \partial g$

$$\begin{aligned} \nabla_{\boldsymbol{x}_i} g =& -\pi_i \boldsymbol{\alpha},\ \boldsymbol{x}_i \in \operatorname{ext} K_1, \\ \nabla_{\boldsymbol{x}_i} g =& \pi_i \boldsymbol{\alpha},\ \boldsymbol{x}_i \in \operatorname{ext} K_2, \\ & \text{for: } \boldsymbol{\alpha} \in \boldsymbol{A}_{\boldsymbol{\alpha}}^{*}, \pi_i \in \Pi_i^{*}, \end{aligned} \tag{3.40}$$

for a dual variable vector $\boldsymbol{\pi} \in \mathbb{R}^{|\operatorname{ext} K_1| + |\operatorname{ext} K_2|}$ with components π_i. To simplify notation, introduce the index sets I_1 and I_2, to indicate the vertex sets of K_1 and K_2 respectively. Whence, the dual feasibility conditions at the optimum of (2.29) are given by

$$\begin{aligned} &\pi_i \geq 0,\ \sum_{i \in I_1} \pi_i = 1,\ \sum_{i \in I_2} \pi_i = 1 \\ &\sum_{i \in I_1} \boldsymbol{x}_i \pi_i - \sum_{i \in I_2} \boldsymbol{x}_i \pi_i = g\hat{\boldsymbol{\beta}}. \end{aligned} \tag{3.41}$$

This implies that $\sum_{i \in I_1} \boldsymbol{x}_i \pi_i$ and $\sum_{i \in I_2} \boldsymbol{x}_i \pi_i$ are points on the convex hull of their respective sets, and are *the same* point when there is no overlap, i.e. when $g = 0$. Thus, a specific contact point on each body can be recovered directly from the solution to the SSH LP.

In the constrained description of rigid body dynamics, the points $\boldsymbol{x}_i$ can be parametrized by the fully redundant configuration $\boldsymbol{q}$, given that the location of each vertex within the body is known in terms of the directors $\boldsymbol{d}_I$. To each vertex $\boldsymbol{x}_i$ associate vector $\boldsymbol{v}_i$ so that the position of the vertex is given by

$$\begin{aligned} \boldsymbol{x}_i &= \boldsymbol{\varphi} + \begin{bmatrix} \boldsymbol{d}_1 & \boldsymbol{d}_2 & \boldsymbol{d}_3 \end{bmatrix} \boldsymbol{v}_i \\ &= \boldsymbol{\varphi} + \boldsymbol{d}_I v_{i,I}. \end{aligned} \tag{3.42}$$

In (3.42), the components of the i^{th} vector, $\boldsymbol{v}_i$, are denoted $(v_{i,1}, v_{i,2}, v_{i,3}) = v_i$. For concreteness let the points in I_1 be associated with a rigid body in configuration $\boldsymbol{q}^1$, and likewise let the points in I_2 be associated with a rigid body configuration $\boldsymbol{q}^2$. By application of the chain rule the relationship

$$\begin{aligned}\nabla_{\boldsymbol{q}^1} g &= \sum_{i \in I_1} \left(\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{q}^1}\right)^T \nabla_{\boldsymbol{x}_i} g \\ \nabla_{\boldsymbol{q}^2} g &= \sum_{i \in I_2} \left(\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{q}^2}\right)^T \nabla_{\boldsymbol{x}_i} g\end{aligned} \tag{3.43}$$

can be recovered. The Jacobian matrix $\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{q}}$ is given by

$$\frac{\partial \boldsymbol{x}_i}{\partial \boldsymbol{q}} = \begin{bmatrix} \boldsymbol{I} & v_{i,1}\boldsymbol{I} & v_{i,2}\boldsymbol{I} & v_{i,3}\boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{3\times 12}. \tag{3.44}$$

Thus, (3.43) can be written as:

$$\begin{aligned}\nabla_{\boldsymbol{q}^1} g &= -\boldsymbol{A}^1 \boldsymbol{\alpha}, \\ \nabla_{\boldsymbol{q}^2} g &= \boldsymbol{A}^2 \boldsymbol{\alpha},\end{aligned} \tag{3.45}$$

with the matrices $\boldsymbol{A}^j$ is given by

$$\boldsymbol{A}^j = \begin{bmatrix} \boldsymbol{I} \\ \sum_{i \in I_j} \pi_i v_{i,1} \boldsymbol{I} \\ \sum_{i \in I_j} \pi_i v_{i,2} \boldsymbol{I} \\ \sum_{i \in I_j} \pi_i v_{i,3} \boldsymbol{I} \end{bmatrix}. \tag{3.46}$$

The separation vector $\boldsymbol{h} = \sum_{i \in I_1} \boldsymbol{x}_i \pi_i - \sum_{i \in I_2} \boldsymbol{x}_i \pi_i$ will also be useful in future sections. This vector can be written as

$$\boldsymbol{h}(\boldsymbol{q}_1, \boldsymbol{q}_2) = \boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2 + \boldsymbol{d}_I^1 \sum_{i \in I_1} v_{i,I} \pi_i - \boldsymbol{d}_I^2 \sum_{i \in I_2} v_{i,I} \pi_i. \tag{3.47}$$

Also, $\nabla_{\boldsymbol{q}^j} \boldsymbol{h}$ is given by $\pm {\boldsymbol{A}^j}^T$, where the $(+)$ corresponds to $j = 1$, and the $(-)$ to $j = 2$.

Making use of the mapping $\boldsymbol{P}^T(\boldsymbol{q}) : T^*_{\boldsymbol{q}}\mathcal{Q} \to T^*_u \mathcal{U}$, the generalized contact forces, $\boldsymbol{\tau}_{\mathcal{A}} = \begin{bmatrix} \boldsymbol{\tau}_\varphi & \boldsymbol{\tau}_T \end{bmatrix}^T = -\boldsymbol{P}^T(\boldsymbol{q}) \partial V_{\mathcal{A}}(\boldsymbol{q})$ with parameter $C \in \mathbb{R}^+$ due to the contact potential of tin Equation (3.38) are given by:

$$\begin{aligned}\boldsymbol{\tau}^1_{\mathcal{A}} &= -gC\boldsymbol{P}^T(\boldsymbol{q}^1)\cdot\boldsymbol{A}_1\boldsymbol{\alpha} = -gC\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{r}^1\times\boldsymbol{\alpha}\end{bmatrix}\\ \boldsymbol{\tau}^2_{\mathcal{A}} &= gC\boldsymbol{P}^T(\boldsymbol{q}^2)\cdot\boldsymbol{A}_2\boldsymbol{\alpha} = gC\begin{bmatrix}\boldsymbol{\alpha}\\ \boldsymbol{r}^2\times\boldsymbol{\alpha}\end{bmatrix},\end{aligned} \tag{3.48}$$

where $\boldsymbol{r}^j = \boldsymbol{d}^j_I \sum_{i\in I_j}\pi_i\boldsymbol{v}_{i,I}$ is a point on the convex hull of each body due to the KKT conditions at the solution of the SSH LP, and, to simplify the notation, $g = g(\boldsymbol{q})$.

3.3.3 Invariance of the Contact Potential.

Before moving on to several illuminating examples for finite element models and rigid bodies, the properties of the SSH LP and its subgradient as they effect the conserved quantities in the systems of interest are investigated. *Noether's theorem* states that if the Lagrangian is invariant with respect to the action of a Lie-group, then a corresponding momentum map is preserved. A complete discussion can be found in [57, 58], however for the simple systems under consideration, a less formal working definition of the theorem may be used. Practically, the action of the Lie group amounts to an invertible, continuously differentiable coordinate transformation. Thus Noether's theorem states that any conserved quantities will remain unchanged under this type of coordinate transformations, i.e. conserved quantities are frame-independent.

The stereotypical example of Noether's theorem in which angular and linear momentum are not necessarily conserved in certain directions is a pendulum ($\boldsymbol{q} = (x, y, z)$) composed of a point mass attached to a massless rigid rod under the action of a gravitational potential

$$V_g = -g\boldsymbol{e}_z^T\boldsymbol{M}\boldsymbol{q}$$

which breaks the the symmetry under $GL(n)$ and $SO(n)$.

In this example, with the reference configuration shown in Fgiure 3.4, the question of momentum conservation amounts to whether a given coordinate transformation leaves the $z-$coordinate of the mass, and therefore the value of V_g, unchanged. For example, translation of the frame along a linear combination of $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ leaves the value of the gravitational potential, and thus the Lagrangian, unchanged. Therefore we expect that the $x-$ and $y-$components of the linear momentum will be conserved. However, translation of the frame along $\boldsymbol{e}_z$ will necessarily alter the value of V_g due to

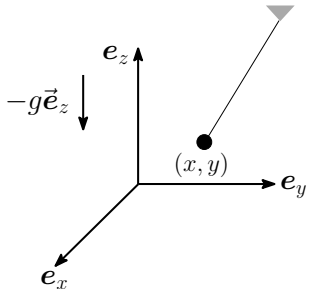


Figure 3.4: Reference configuration for a pendulum composed of a point mass connected to a rigid massless rod in 3d.

$z \neq \bar{z}$, so $z-$component of linear momentum will not be conserved. This is illustrated in Figure 3.5.

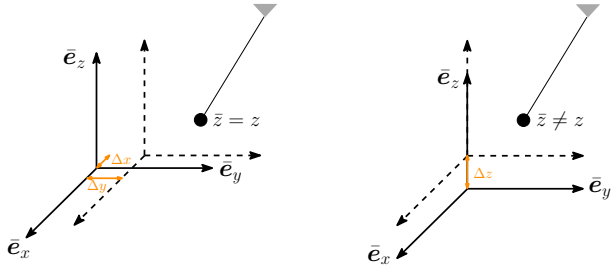


Figure 3.5: In the pendulum example, translation of the reference frame in the $x - y$ (left) plane renders $\bar{z} = z$, but this is not true for translations along $\boldsymbol{e}_z$ (right) for which $\bar{z} \neq z$.

Furthermore, *rotations* of the frame about either $\boldsymbol{e}_x$ or $\boldsymbol{e}_y$ result in a change in the vale of the $z-$coordinate of the point mass, which implies that the $x-$ and $y-$components of the system's angular momentum are, in general, not conserved. However, all rotations about $\boldsymbol{e}_z$ render $z = \bar{z}$, so the $z-$component of the angular momentum will be conserved. This is illustrated in Figure 3.6.

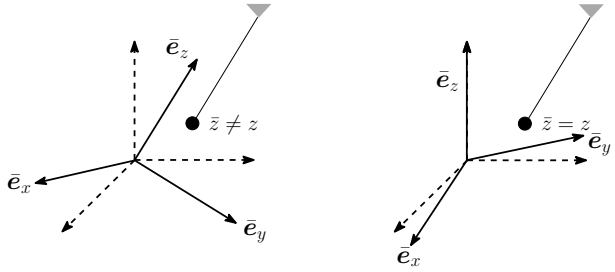


Figure 3.6: In the pendulum example, rotation of the referenec frame about $\boldsymbol{e}_x$ or $\boldsymbol{e}_y$ (left) gives $\bar{z} \neq z$, but this is not true for rotations about $\boldsymbol{e}_z$ (right) for which $\bar{z} = z$.

By way of example in the case of an SSH LP based contact potential, consider a simple system of two rigid bodies in two dimensions, so that the configuration is described by the generalized coordinates $\boldsymbol{q} = \left(x^1, y^1, \theta^1, x^2, y^2, \theta^2\right)$, with the angular degrees of freedom described in relation to the $x-$axis.

Without loss of generality, we take a reference configuration with one body centered at the origin, and the other body situated so that $V_{\mathcal{A}}(\boldsymbol{q}) = 0$, with $\theta^1 = \theta^2 = 0$. An example of one such system is shown in Figure 3.7. Recall that the SSH LP approximates a quadratic program to determine the (scalar) signed distance between bodies. Clearly, this function is invariant under the action of $GL(n)$ and $SO(n)$, however the invariance of the *linear* program depends on the choice of the vector, $\hat{\boldsymbol{\beta}}$. By requiring $\hat{\boldsymbol{\beta}}(\boldsymbol{q}^1, \boldsymbol{q}^2)$ such that the norm $\hat{\boldsymbol{\beta}}^T \boldsymbol{d}_I^{j*}$ for the basis vectors $\boldsymbol{d}_I^j$ of each body's frame, the invariance of the SSH LP is maintained.

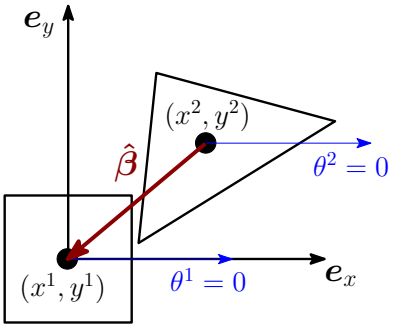


Figure 3.7: Arbitrary configuration of a system of two polygonal rigid bodies in 2d.

Linear Momentum. Conservation of linear momentum corresponds to translational symmetry of the Lagrangian. Clearly, the signed distance function between two bodies does not depend on the location of the origin of the system (see Figure 3.8), rendering the contact potential invariant to the action of $GL(n)$.

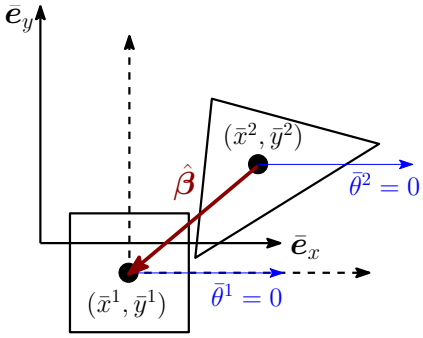


Figure 3.8: Invariance of the signed distance function w.r.t. translation of the reference frame.

Angular Momentum. Conservation of angular momentum corresponds to rotational symmetry of the Lagrangian. That is, for a (fixed) arbitrary choice of the origin of the reference frame, if the Lagrangian remains invariant to rotations of the frame about a given axis, $\boldsymbol{n}$, then angular momen-

*This requirement makes use of the fact that the directors used as redundant degrees of freedom in the constrained integrator form a spatial basis.

tum is conserved about that axis. By selecting $\hat{\boldsymbol{\beta}}$ which only depends on the relatively translation of the bodies, the invariance of the SSH LP is also preserved, as shown in Figure 3.9.

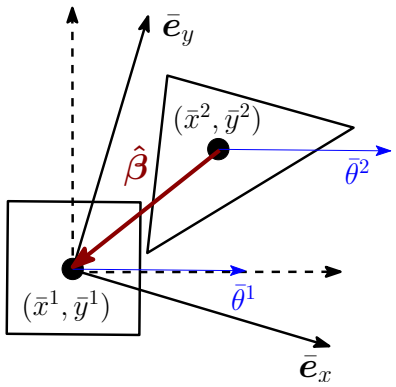


Figure 3.9: Invariance of the signed distance function w.r.t. rotation of the reference frame.

So, although some numerical errors can be expected, the use of the SSH LP directly as a contact potential will not effect the conservation of the otherwise conserved components of the angular or linear momentum. For finite elements, $\hat{\boldsymbol{\beta}}$ is given by (2.69), and for rigid bodies

$$\hat{\boldsymbol{\beta}} = \frac{\boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2}{\|\boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2\|}$$

is used. This requires that the gradient be appropriately amended by the chain rule by augmenting the matrices $\boldsymbol{A}^j$ to

$$\begin{aligned} \overline{\boldsymbol{A}}^1 &= \boldsymbol{A}^1 + \boldsymbol{A}_\beta \\ \overline{\boldsymbol{A}}^2 &= \boldsymbol{A}^2 - \boldsymbol{A}_\beta \end{aligned} \tag{3.49}$$

with

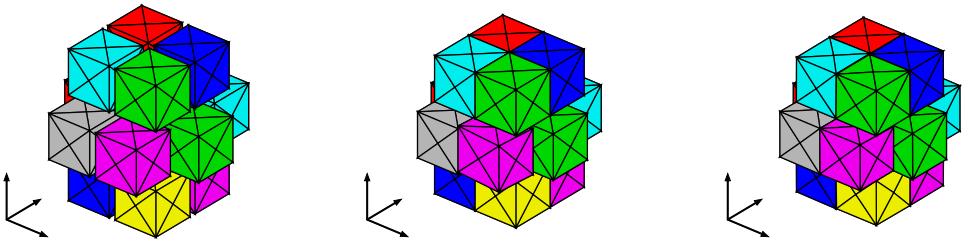
$$\boldsymbol{A}_\beta = g(\boldsymbol{q}) \begin{bmatrix} \frac{1}{\|\boldsymbol{\varphi}_\Delta\|} \left(\boldsymbol{I} - \frac{1}{\|\boldsymbol{\varphi}_\Delta\|^2} \boldsymbol{\varphi}_\Delta \otimes \boldsymbol{\varphi}_\Delta \right) \\ \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} \end{bmatrix} \tag{3.50}$$

in which $\boldsymbol{\varphi}_\Delta = \boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2$ and it is noted that the value of the SSH LP at the solution is equivalent to the dual variable associated with the inequality constraint.

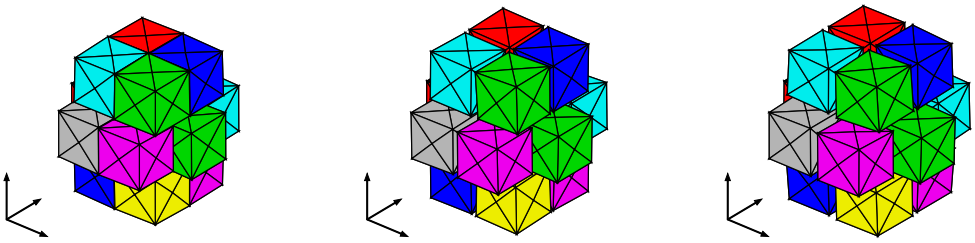
3.3.4 Examples: Explicit Dynamics with Newmark-β

The following examples use a simple explicit Newmark time-integration scheme with a mid-point rule for the velocity ($\beta = \frac{1}{2}$) calculation and a contact potential, as discussed in the introduction. The subgradient of the modified SSH LP (2.68), multiplied by a constant factor, is used directly to determine contact forces in both rigid body and finite element simulations. These examples emphasize that the use of the SSH linear program directly as an approximation to the indicator function allows us to successfully model complicated collisions, non-convex geometries, and extremely stiff systems using only the most basic explicit time-integration scheme with a relatively large time-step.

Cluster of Cubes. A cluster of 16 cubes made of a neo-Hookean material ($\lambda = 10e3$ Pa, $\mu = \lambda/2$), and modeled using lumped mass tetrahedral conforming solid elements with first order interpolation is considered next, and the time step in this example is $h = 10e-4$ seconds. All cubes start with an inward-radial velocity towards the cube in the center of the cluster proportional to the distance between each body's center of mass and the origin, and no angular velocity. This is done specifically to induce collision configurations with primal and dual degeneracies. The time lapse images in Figure 3.10 illustrate qualitatively the preservation of symmetry in the pre- and post-collision trajectories of the cubes. Figure 3.11 shows the energy and momentum behavior of the entire system of bodies in the simulation. The total energy, and the translational and angular momenta are preserved for the lumped-mass model.

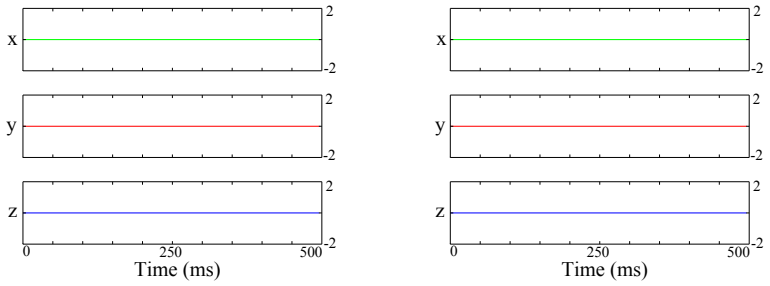


(a) $t = 0.0s$ (b) $t = 0.084s$ (c) $t = 0.108s$

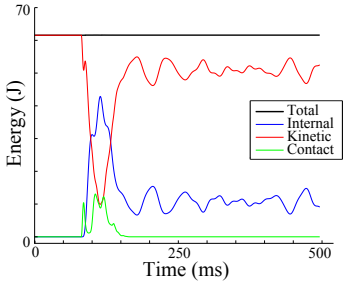


(d) $t = 0.132s$ (e) $t = 0.216s$ (f) $t = 0.258s$

Figure 3.10: Collisions of Neo-Hookean (λ = 10e3 Pa, $\mu = \lambda/2$) cubes with no external forces or potentials, h = 10e $-$4s. The mesh used in the simulation is shown in the figures. The first set of (simultaneous) collisions occurs at $t = 0.084s$ between the cubes in the top, bottom, and middle layer, respectively. Several milliseconds later, the cubes in the top and bottom layer strike the cubes in the middle layer. There are about 45ms of sustained contact before the cubes disperse.



(a) Translational momentum (b) Angular momentum



(c) Energy

Figure 3.11: Momentum and energy histories for neo-Hookean cube lumped-mass finite element simulation.

Kinematic 'Satellite' Docking. The following example also makes use of lumped-mass tetrahedral conforming linear solid elements which model a relatively stiff ($\lambda = 10e4$ Pa, $\mu = \lambda/2$) neo-Hookean material using $h = 10e-5$s. In this coarse model, two identical bodies are used, each with a set of pins, and a complementary set of holes. One such body is shown in Figure 3.12. As can be seen in Figures 3.12 and 3.13, the diameter of the bulge in the forward portion of the pins and of the cuff at the base of the pins is larger than the smallest diameter of the holes, so that both pins and holes must deform in order for latching to take place. The distance between the cuff and the base of the pin is slightly larger than the thickness of the shell. Figure 3.15 shows several configurations of the bodies as time advances, and Figure 3.14 shows the energy and momentum behavior of the two body system over the course of the simulation. Both of these figures show the sustained contact between the bodies once latching has taken place. Furthermore, note that the body on the right, which has a velocity of magnitude 3.5m/s at $t = 0$s in the negative x-direction, is initially off-set in the $y-$direction from the body on the left, which is initially at rest, so that the conical holes must guide the pins into alignment. A closeup of this alignment is shown in Figure 3.13.

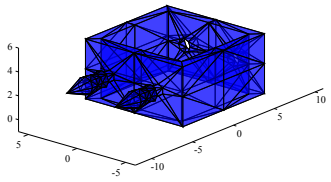


Figure 3.12: Geometry and mesh of bodies used in the kinematic latching simulation.

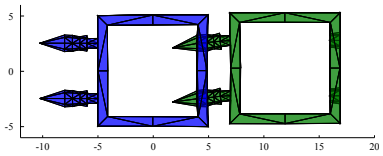
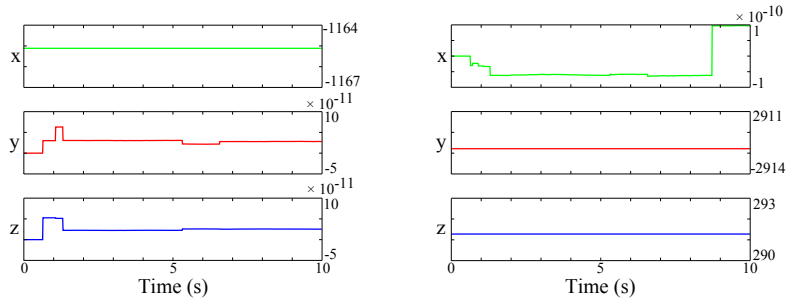
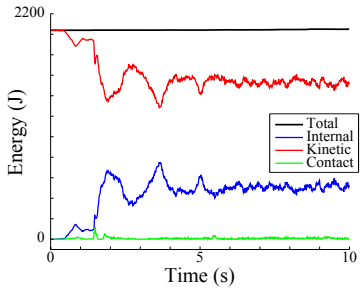


Figure 3.13: Transparent top-down view of pins being guided into conical holes at $t = 0.75s$



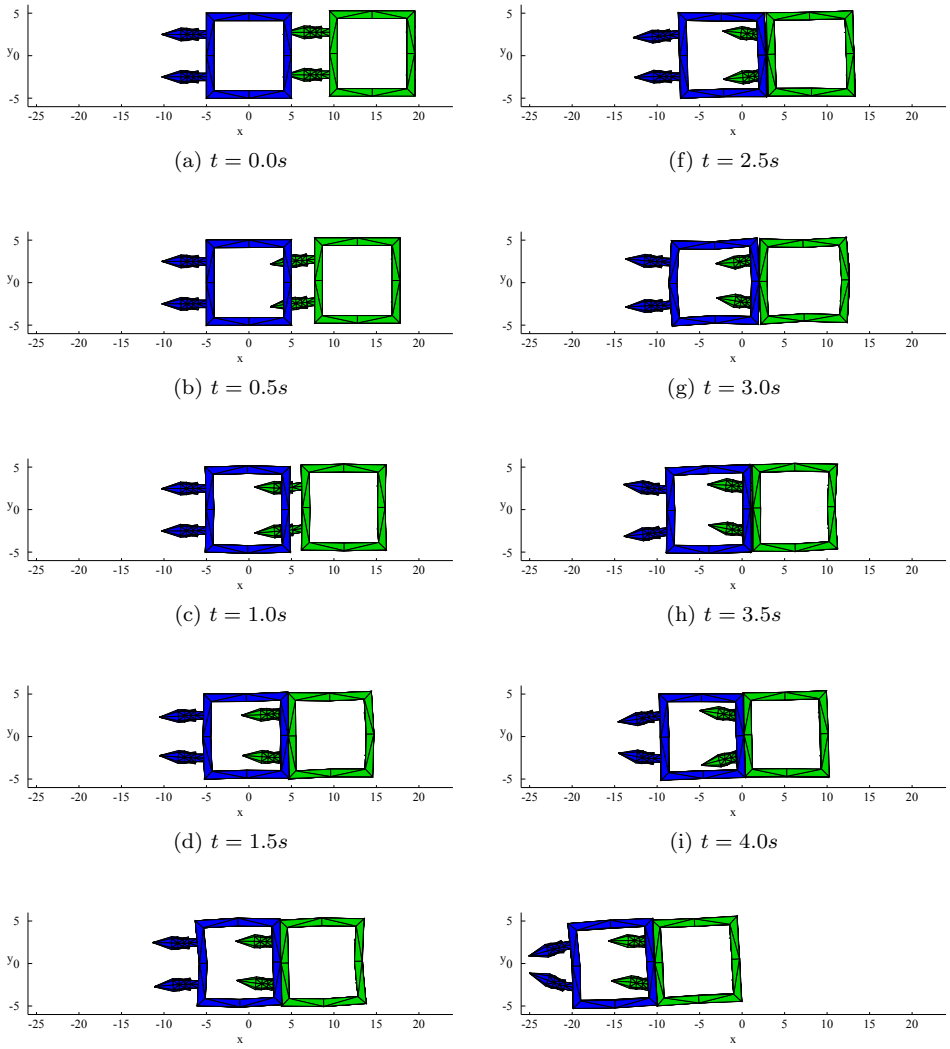
(a) Translational momentum

(b) Angular momentum



(c) Energy

Figure 3.14: Momentum and energy histories for Neo-Hookean 'satellite' latching simulation.



(a) $t = 0.0s$

(b) $t = 0.5s$

(c) $t = 1.0s$

(d) $t = 1.5s$

(e) $t = 2.0s$

(f) $t = 2.5s$

(g) $t = 3.0s$

(h) $t = 3.5s$

(i) $t = 4.0s$

(j) $t = 10.0s$

Figure 3.15: Time lapse images of kinematic 'satellite' latching simulation.

Clumping Rigid Hexagons. Each of the following two examples uses a group of 16 identical hexagonal rigid bodies. Collisions are modeled as frictionless and inelastic, with restitution modeled by the introduction of a non-holonomic force which penalizes the normal velocity when the bodies overlap, akin to a drag force on the relative normal components of the velocities. For both examples, the initial positions and velocities are the same, as shown in Figure 3.16, where the relative magnitude and direction of the translational velocity is shown, and each body also has a random initial angular velocity, and the time step is $h = 10e{-}3$ seconds.

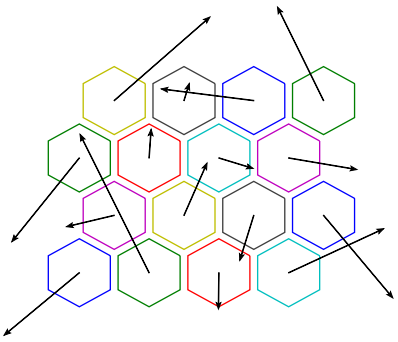
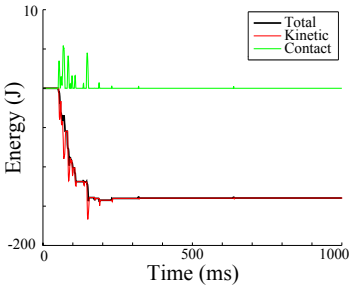
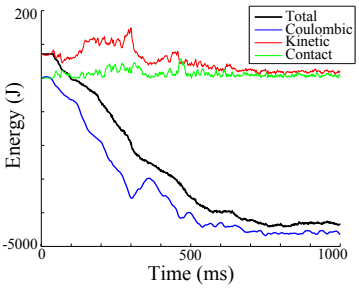


Figure 3.16: Initial configuration and translational velocities for rigid hexagon examples. All bodies also have a random initial angular velocity (not shown).

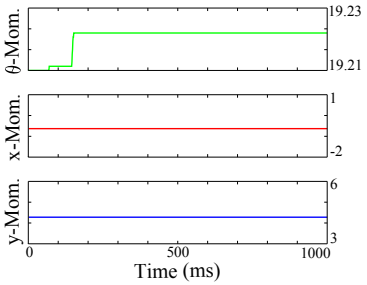
In the first example, there are no potentials aside from the contact potential given by the SSH LP. After several collisions, occasionally with multiple collisions in one time step, the bodies disperse, see Figure 3.18. In the second example, each identical hexagonal body is equipped with alternating positive and negative Coulombic charges centered on each face and off-set slightly from the surface to avoid a singularity in the calculation of the Coulombic potential. Due to the strong attractive potential, the bodies lock together into a regular grid, see Figure 3.19. Note that the second system, in particular, becomes extremely stiff as the bodies lock together, and the SSH LP contact potential successfully allows the bodies to oscillate around a minimum Coulombic potential energy, as illustrated in Figure 3.17.



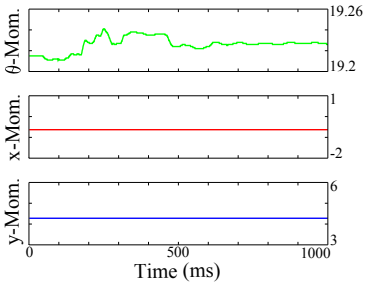
(a) Energy, only contact potential



(b) Energy, Coulombic potential centered on faces

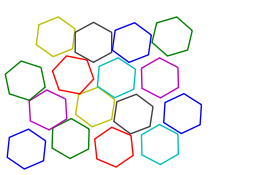


(c) Momentum, only contact potential

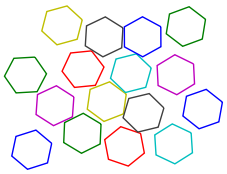


(d) Momentum, Coulombic potential centered on faces

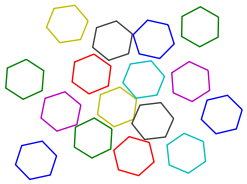
Figure 3.17: Energy and momentum histories for rigid hexagons undergoing frictionless, inelastic collisions using the SSH LP as a contact potential. In the first example, the initial kinetic energy in the system is taken as the baseline (zero) energy, and in the second example, the initial Coulombic potential energy is taken as the baseline (zero) energy. In both cases, the loss of total energy in the system is due to inelastic collisions and a small amount of numerical dissipation associated with the Newmark algorithm.



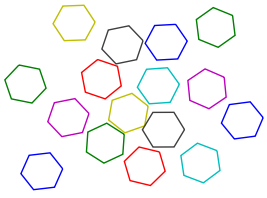
(a) $t = 0.125s$



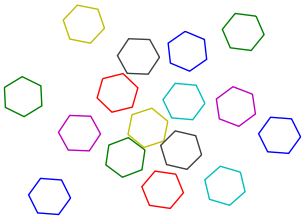
(b) $t = 0.250s$



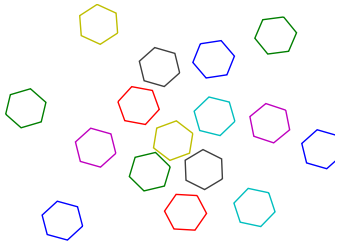
(c) $t = 0.375s$



(d) $t = 0.500s$

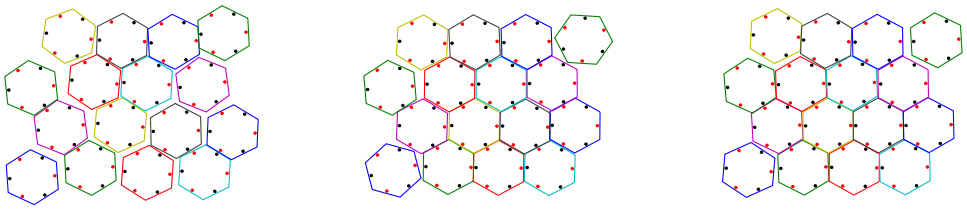


(e) $t = 0.750s$

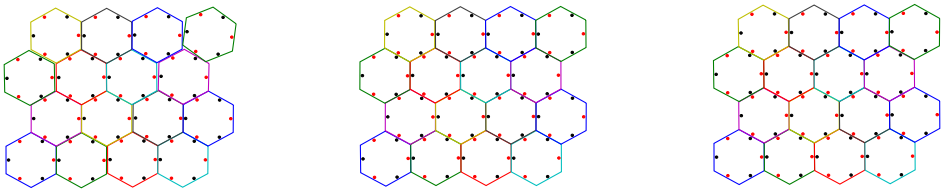


(f) $t = 1.0s$

Figure 3.18: Frictionless inelastic collisions leading to the dispersal of hexagonal rigid bodies. The initial configuration is shown in Figure 3.16.



(a) $t = 0.125s$ (b) $t = 0.250s$ (c) $t = 0.375s$



(d) $t = 0.500s$ (e) $t = 0.750s$ (f) $t = 1.0s$

Figure 3.19: Clumping of rigid bodies due to Coulombic forces within the bodies at positions indicated by dots. Collisions are frictionless and inelastic. The initial configuration is shown in Figure 3.16.

3.3.5 Examples: Constrained Mechanical Systems

In the following examples, the constrained mechanical integrator described in [50] and outlined at the beginning of this chapter is used directly with a smoothed contact potential to model dynamics with elastic collisions between rigid bodies. The potential, $V_{\mathcal{A}}(\boldsymbol{q})$, is derived directly from the SSH LP, as described in Chapter 2. Unless otherwise stated, the energy plotted is associated with the discrete Hamiltonian, H_d, given by

$$H_d(\boldsymbol{q}_k, \boldsymbol{p}_k) = \frac{1}{2}\boldsymbol{p}_k^T \boldsymbol{M}^{-1}\boldsymbol{p}_k + \boldsymbol{V}(\boldsymbol{q}_k),$$

with the momenta p_k given by the constrained discrete Legendre transform. The corresponding total angular momentum vector $\boldsymbol{L}$ can be recovered from the redundant momenta and configuration as

$$\boldsymbol{L} = \boldsymbol{\varphi} \times \boldsymbol{p}_{\varphi} + \boldsymbol{d}_I \times \boldsymbol{p}_I,$$

in which the repeated index I indicates summation. Labels of 'Energy' and 'Angular Momentum' or $\boldsymbol{L}$ refer to these quantities unless otherwise noted.

'Newton's Cradle'. This example is intended as a 'gut-check' for the accuracy of $\partial V_{\mathcal{A}}(\boldsymbol{q})$ based on the SSH LP for non-smooth bodies, and is motivated by the dynamics of the well-known Newton's Cradle toy (see Figure 3.20) modified so that the balls are replaced by cubes, and the motion of the bodies is unconstrained (i.e., the strings are not modeled so there is only one series of collisions).

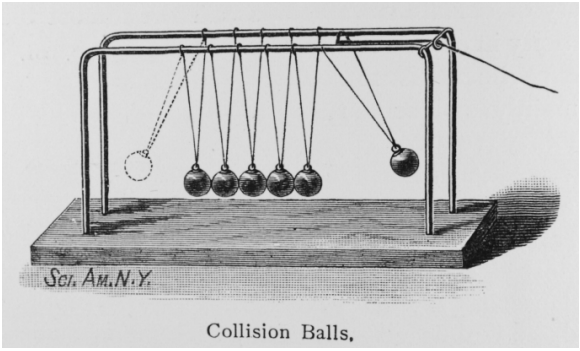
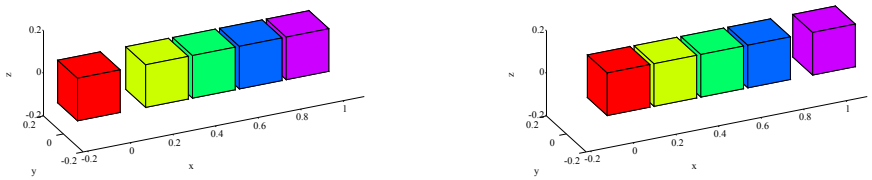


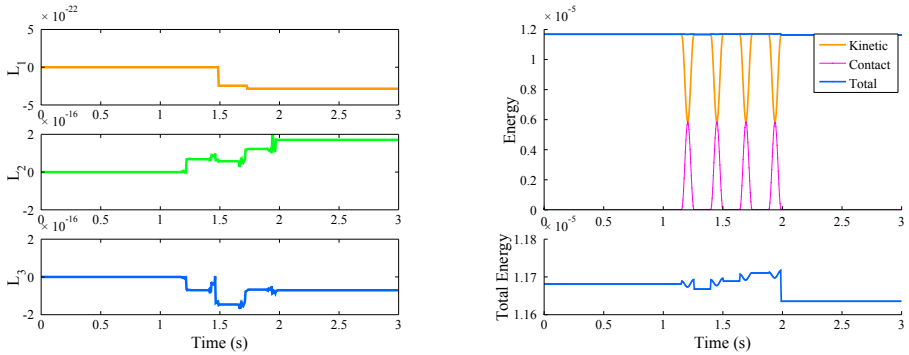
Figure 3.20: Sketch of Newton's cradle toy with 5 balls (source: D. Simanek, http://www.lhup.edu/ dsimanek/scenario/cradle.htm).

Two cases are considered. In case 1, four cubes are aligned at rest of side length $l = 0.2\sqrt{2}$ m and uniform density $\rho = 270$ kg/m^3. They are separated by $l/10$. One cube of the same dimension starts with its face a distance of $l/2$ from the face of the end cube and moves toward the bunch with initial velocity $\dot{\boldsymbol{\varphi}}^1 = [0.105\ 0\ 0]^T$ m/s. As can be seen in the images in Figure 3.21, the impulse of the initial impact leads to the cube on the opposite end of the bunch from the original moving cube departing the cluster. The evolution of the energy and angular momentum for this case are shown in figure 3.22. In this example, the contact potential parameter is $C = 1$, and the time step is $h = 0.005$s. The choice of time-step and contact potential parameter is empirical, and calls for a trade-off between accuracy–with higher accuracy achieved by decreasing h and increasing C–and efficiency–with faster run times achieved by increasing h, which requires a corresponding decrease in C.



(a) t = 0s (b) t = 3s

Figure 3.21: Initial and final configurations for first case of 'Newton's Cradle' examples.



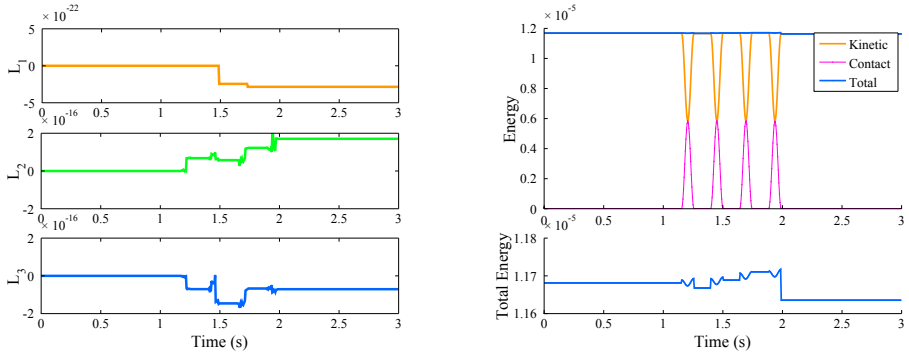
(a) Angular Momentum. (b) Energy.

Figure 3.22: Angular momentum and energy evolution for and first case of 'Newton's Cradle' examples.

In case 2, three cubes are aligned at rest with a side length $l = 0.2\sqrt{2}$ m and uniform density $\rho = 270$ kg/m^3 with separated by $l/100$. Two cubes of the same dimension start with faces a distance of $l/2$ from the face of the end cubes and move toward the bunch with initial velocities $\dot{\boldsymbol{\varphi}}^1 = [0.105\ 0\ 0]^T$ m/s and $\dot{\boldsymbol{\varphi}}^5 = [-0.105\ 0\ 0]^T$ m/s, respectively. As can be seen in the images in Figure 3.23, the impulse of the initial impact leads to both of the end cubes moving away from the cluster. The evolution of the energy and angular momentum for this case are shown in Figure 3.24. In this example, the contact potential parameter is $C = 1$, and the time step is $h = 0.005$s.

(a) t = 0s (b) t = 2.2s

Figure 3.23: Initial and final configurations for the second case of 'Newton's Cradle' examples.



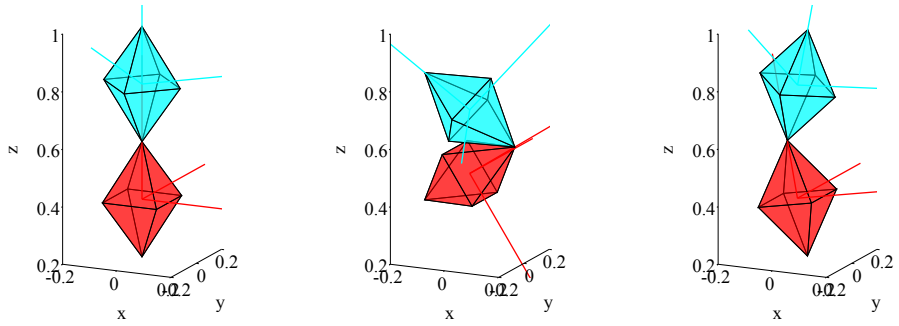
(a) Angular Momentum. (b) Energy.

Figure 3.24: Angular momentum and energy evolution for and second case of 'Newton's Cradle' examples.

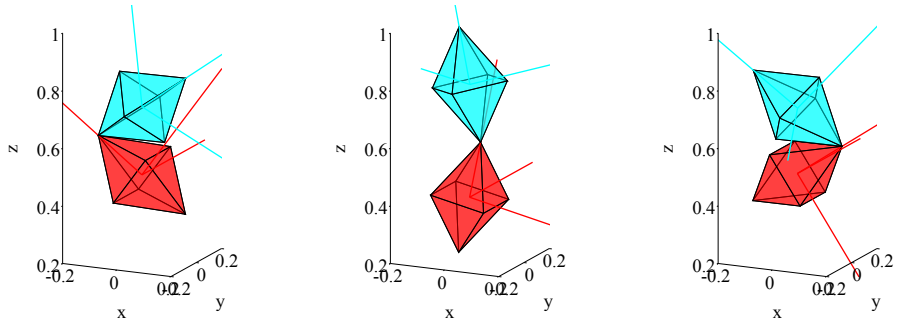
In both of these examples, no relative rotation between the bodies is induced by the use of a the SSH LP contact potential, which is the expected result for perfectly flat face-face contact. The small changes in the energy in due to the presence of the potential in the system are consistent with previous results, which show that the discrete Hamiltonian may not be exactly preserved in such cases (c.f. [50]).

While the contact potential generates a conservative force field, relatively large time steps do not accurately capture this fact because the potential is either 'on' or 'off' for the entire time step–depending on the midpoint configuration–even though in the continuous case it varies smoothly along the path. The overall energy behavior can be improved by reducing the time step or softening the contact potential to further smooth the variation, which will be verified in the next example.

Metronome. In this example, a simple kinematic chain consisting of two octohedral prisms with side length $0.2\sqrt{3}$ m connected by a spherical joint at the top of the first body and the bottom of the second body is introduced. In the initial configuration, $\boldsymbol{d}_3^1 = \boldsymbol{d}_3^2$, and the remaining directors of the second body are rotated by $\frac{\pi}{4}$rad about $\boldsymbol{d}_3$ relative to the first. The initial velocities used to start the simulation are $\boldsymbol{\omega}_1 = [0\ 1\ 0]^T$ rad⁄s and $\boldsymbol{\omega}_2 = [0\ -1\ 0]^T$ rad⁄s, so that the bodies rotate about the joint and swing towards each other, collide when the edge of the second body strikes the face of the first body, and so on. Several time lapse images of this motion are shown in Figure 3.25. The evolution of the energies and angular momentum for the potential-based metronome example are show in Figure 3.26. In the present case, $C = 1$ and $h = 0.0015$s.



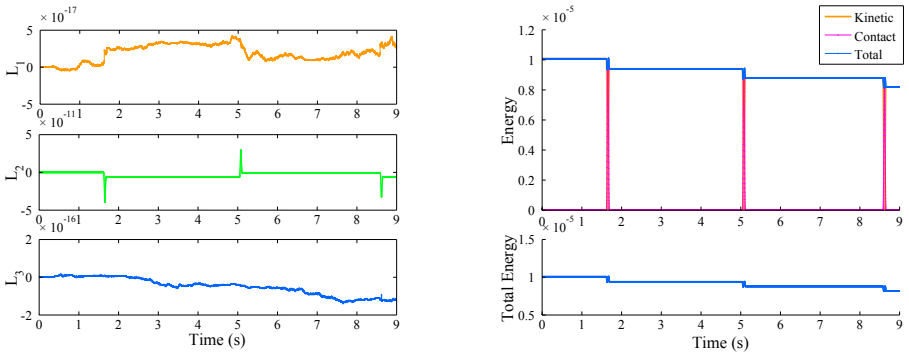
(a) $t = 0.0$s (b) $t = 1.8$s (c) $t = 3.6$s



(d) $t = 5.4$s (e) $t = 7.2$s (f) $t = 0.1$s

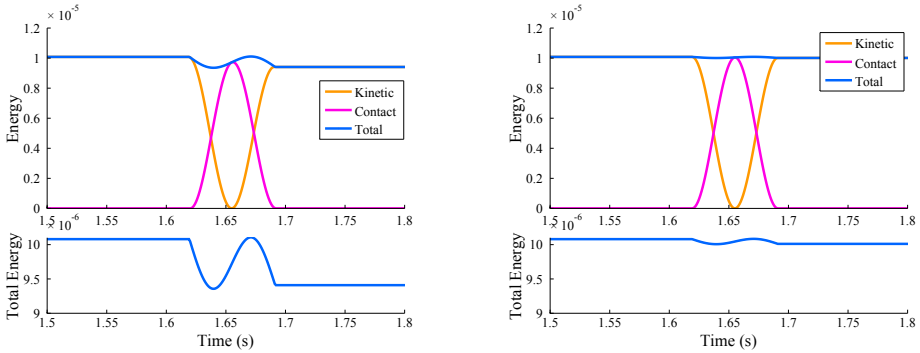
Figure 3.25: Time lapse images for a prismatic metronome connected by a spherical joint with collisions modeled using a smooth SSH LP=based contact potential.

As in the Newton's cradle examples, no additional relative rotation is induced by the use of the SSH LP contact potential, which is the expected result for this particular edge-face contact configuration. However, as discussed in the previous section, the presence of the stiff unilateral contact potential affects the overall energy of the system. This behavior can be improved either through the reduction of the contact parameter C, at the cost of a loss in accuracy of the configuration due to increased non-physical overlap, or a reduction in the time step h. Figure 3.27 shows the effect of reducing the time step to $h = 1e - 4$s on the total energy.



(a) Angular Momentum. (b) Energy.

Figure 3.26: Angular momentum and energy evolution for a prismatic metronome connected by a spherical joint.



(a) Energy, $h = 0.0015s$. (b) Energy, $h = 0.0001s$.

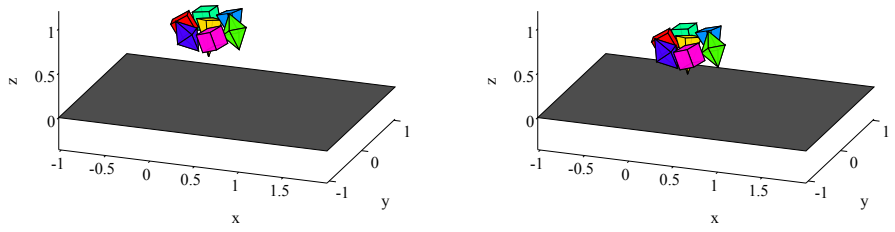
Figure 3.27: Evolution of the total energy of the prismatic metronome for $h = 1.5e-3$s and $h = 1.0e-4$s, respectively. The total energy plotted in the bottom plot is on the same scale for both examples.

Falling Polyhedra. In this example, a gravitational potential given by:

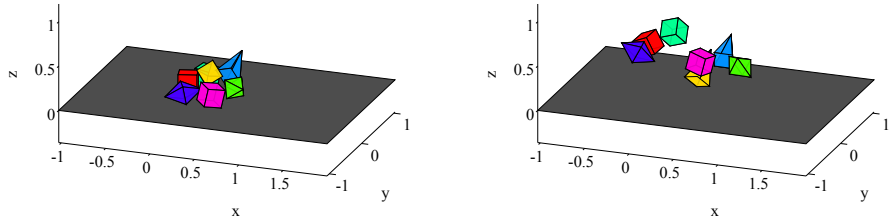
$$V_g(\boldsymbol{q}) = -g\boldsymbol{e}_g^T \boldsymbol{M}\boldsymbol{q}$$

is introduced into a system of seven polyhedra, with $g = 9.81\text{m}/\text{s}^2$ and $\boldsymbol{e}_g^i = [\boldsymbol{e}_z\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}]^T \in \mathbb{R}^{12}$. The polyhedra start in a flower-like cluster with centers of mass ~ 0.8m above a 'floor' occupying the $x-y$ plane, with a small $\sim 0.01\text{m}/\text{s}$ random initial translational velocity, and no initial angular velocity. For the purposes of this example, the floor is also modeled as a polyhedron, with vertices at $(x, y) = (\pm 10, \pm 10)$cm, and normal $\boldsymbol{e}_z$. In the evaluation of the SSH LP for body-floor interactions, $\hat{\boldsymbol{\beta}} = \boldsymbol{e}_z$ is always used. Three different shapes are used; a cube with side length 0.2m, a right square pyramid with base dimension 0.2m and height 0.4m, and a regular octohedral prism with side length $0.2\sqrt{3}$m, all with a uniform density of $\rho = 270\text{kg}/\text{m}^3$. A series of time lapse images of the system is shown in Figure 3.28, and the evolution of the total energy and angular momentum is shown in Figure 3.29 for $C = 10$ and $h = 5e-4$. Again, the choice of h and C is largely empirical, however the pressence of the gravitational potential which opposes the contact potential in body-floor collisions necessitates a larger choice of C to maintain reasonable physical fidelity than the previous examples in which no potentials other than the contact potential were active. Figure 3.30 shows an example inadmissible configuration, which is allowed by this approach.

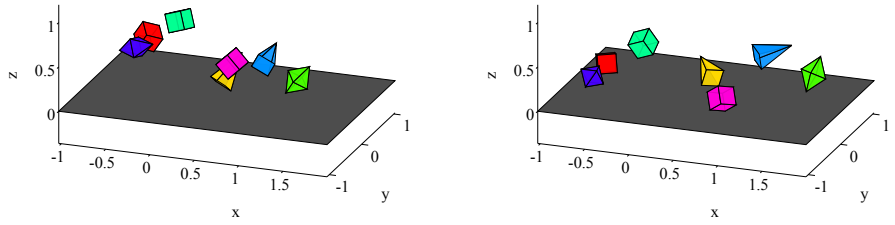
In this example, the bodies do not collide with each other prior to colliding with the floor. Once the bodies do hit the floor, rotation is induced and a series of body-body and body-floor collisions take place in quick succession. Due to rotational symmetry of the system about the $z-$axis, the angular momentum about the $z-$axis, L_3, is preserved in the presence of both the gravitational and contact potentials. The changes in the total energy of the system seen in Figure 3.29 can be attributed to the choice of C and h, where a trade off between accuracy and efficiency must be made.



(a) $t = 0$s (b) $t = 0.2$s

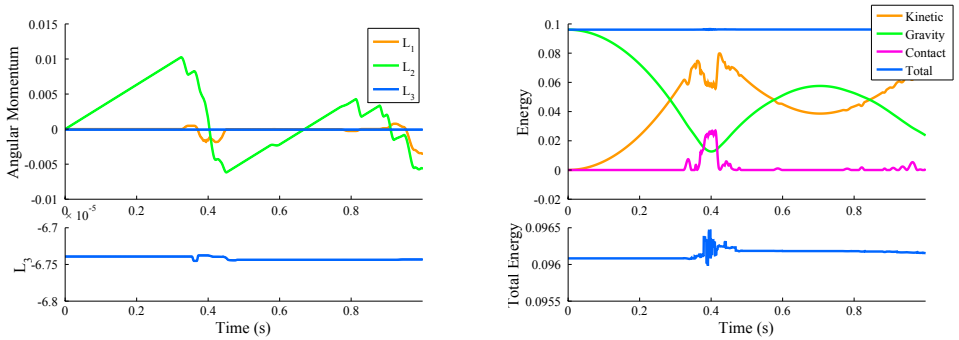


(c) $t = 0.4$s (d) $t = 0.6$s



(e) $t = 0.8$s (f) $t = 1.0$s

Figure 3.28: Time lapse images for falling polyhedra using the contact potential approach base on the SSH LP.



(a) Angular Momentum, $h = 5e-4$s, $C = 10$. (b) Energy, $h = 5e-4$s, $C = 10$.

Figure 3.29: Angular momentum and energy evolution for falling polyhedra with $C = 10$ and $h = 5e-4$s. Changes in $\boldsymbol{L}$ are due to the presence of gravity or to body-floor collisions. The overall energy behavior could be improved by decreasing either C, h, or both.

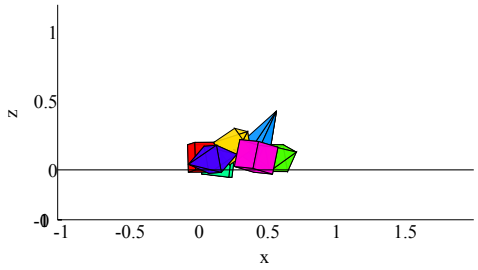


Figure 3.30: Example of allowed overlap for several body-floor collisions at time $t = 0.4$s

Hexagonal Prisms. As a final example of the potential based variational approach, a system of (potentially) clumping rigid hexagonal prisms is treated. The hexagons have height 0.4m and a side length of 0.2m. As in the two-dimensional example–which differs from the present example due to the use of the Newmark integration scheme–a series of alternating positive and negative Coulombic charges are centered behind each side face, with the Coulombic potential for a single interaction given by:

$$V_c^i(\boldsymbol{q}) = \frac{Kc_1c_2}{\sqrt{\boldsymbol{d}_c^{i\,T}\boldsymbol{d}_c^i}} \qquad (3.51)$$
$$\boldsymbol{d}_c^i = \boldsymbol{\varphi}^{1,i} - \boldsymbol{\varphi}^{2,i} + \boldsymbol{X}^{1,i}\boldsymbol{r}_c^{1,i} - \boldsymbol{X}^{2,i}\boldsymbol{r}_c^{2,i},$$

with $\boldsymbol{r}_c^{j,i}$ given in terms of the directors for the j^{th} body, $c_1, c_2 \in \{-1, 1\}$ are the sign of the charge, and $K \in \mathbb{R}$ is a constant. In the present example, $K = 5e-4$. Due the presence of the Coulombic potential, the contact parameter C must be carefully tuned to prevent the calculation of the Coulombic potential in a singular or near-singular configuration. The distance from the center of mass to the charges is $\|r_{cou}\| = 0.1$m to prevent the charges from overlapping in an achievable overlap configuration as can be seen in Figure 3.31.

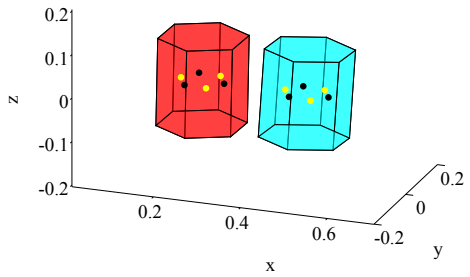
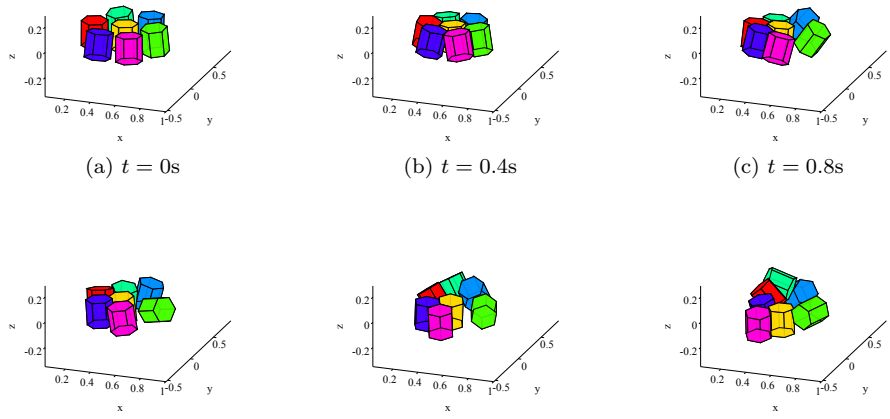


Figure 3.31: Hexagonal prisms shown with locations of alternating positive and negative (yellow and black) Coulombic charges.

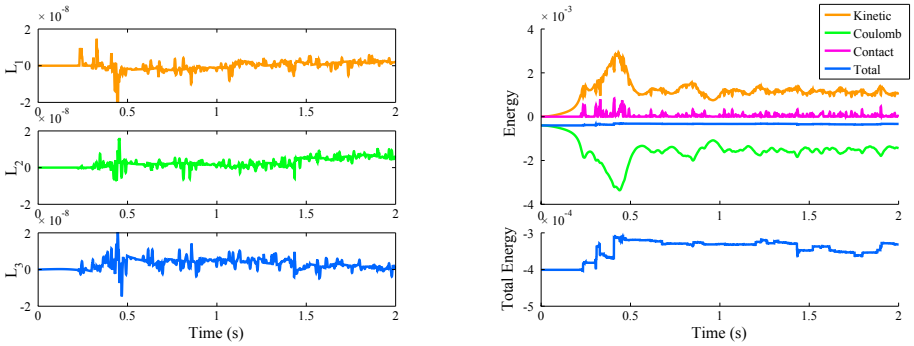
As in the previous examples, a trade-off must be made between accuracy and efficiency. That is, the contact parameter must be chosen so that the contact forces are stronger than Coulombic forces at close range, but, as we have seen, this requires a corresponding decrease in the time step. The results for the choice of $C = 50$ and $h = 5e-4$s are shown below, with the evolution of the configuration in Figure 3.32 and the evolution of the energy shown in Figure 3.33.



(a) $t = 0$s (b) $t = 0.4$s (c) $t = 0.8$s

(d) $t = 1.2$s (e) $t = 1.6$s (f) $t = 2.0$s

Figure 3.32: Time lapse images for hexagonal prisms with embedded Coulombic charges (not shown, see Figure 3.31) the contact potential approach based on the SSH LP using the constrained discrete variational integrator.



(a) Angular Momentum. (b) Energy.

Figure 3.33: Angular momentum and energy for hexagonal prisms with embedded Coulombic charges, contact parameter $C = 50$, and time step $h = 5e - 4$.

3.3.6 Discussion

The direct use of a contact potential based on the SSH LP to model elastic collisions between non-smooth polyhedral bodies is particularly useful if the inaccuracy in the motion due to the allowed overlap is tolerable for a given simulation. The accuracy of the SSH LP-based contact potential in terms of $\partial V_{\mathcal{A}}(\boldsymbol{q}) \approx N_{\mathcal{A}}(\boldsymbol{q})$ leads to realistic simulations and allows for clumping and multiple collisions to be modeled without concern about blow-up in the modeled energy due to spurious motions or deformations. This is a direct result of the solution structure to the SSH LP, which leads to *normal* contact forces which are *local* to the features involved in the collisions. This property of the subgradient is unique to the linear programming interpenetration function developed in Chapter

2. Furthermore the invariance of the potential under the action of the Lie groups outlined in Section 3.3.2 means that its use does not affect quantitie (i.e. linear and angular momentum components) that are expected to be conserved in a given system.

The limitations on this method are largely imposed by the need for accuracy in the sense of allowable overlap and energy conservation. To reduce the overlap, the contact parameter C must be increased to stiffen the potential. However, to maintain good conditioning and reasonable conservation of energy for the variational integrator, any increase in C must be accompanied by a corresponding decrease in h. This conflict also arises when other potentials are present in the system which drive the system towards collisions. It should be possible to improve this behavior by, e.g., employing one of the time-adaptive methods developed in Moore [60], Moore et al. [61] to reduce to time step only in the vicinity of the stiff contact potential.

In conclusion, the approximation of elastic collisions between polyhedral bodies via a parameter-dependent contact potential based on the SSH LP is a straightforward way to treat the equations of motion in a non-smooth setting if the loss of accuracy in the motion and conserved quantities and due to overlap is tolerable for a given simulation.

3.4 Exact Contact Dynamics

In this section we dispense with the potential approximation for elastic contact for constrained mechanical systems in favor of an exact approach which closely follows Fetecau et al. [25] and Leyendecker et al. [53]. The essence of the method is to introduce the physical contact time(s), t_c, as an additional variable in the equations of motion. To begin, the action integral and corresponding Euler-Lagrange equations are developed for constrained systems in a continuous setting. With the contact time as a variable, it is no longer necessary to consider a system involving the indicator function. Rather, the equations of motion at the contact time become a set of jump conditions on the energy and momenta of the system–which, as shown in [25], have a one-to-one correspondence to the *exact* result in the non-smooth setting*. The same is true in a discrete setting, in which variations in the configuration at the contact time have the additional restriction that $\delta \boldsymbol{q}_c \in T\mathcal{C} \cap T\partial\mathcal{A}$, leading to constraints on the discrete analogs to energy and momentum.

While the integrator presented in this section does follow [25] and [53] quite closely, a subtle modification is made to the derivation of the continuous jump conditions for constrained mechanical systems to make the description of collisions in this type of system more precise. In addition to demonstrating the application of the method to non-smooth bodies enabled by the SSH LP, the relationship between discrete energy jump conditions on the configuration and momentum levels (i.e., in terms of E_d and H_d respectively) is numerically investigated the conditions are shown to be equivalent, with slight advantages in computational time for the latter (momentum-level) condition. This latter investigation is useful in Section 3.3 for the formulation of a decomposition-based approximation to the present exact variational collision integrator.

3.4.1 Continuous Constrained Equations of Motion

In this section, the impact equations are developed in a continuous constrained Lagrangian setting (c.f. [15, 25, 53]), with the constraint manifold and its tangent bundle given by (3.5) and (3.6).

Note that $\mathcal{C}$ can be embedded in $\mathcal{Q}$ via $i : \mathcal{C} \to \mathcal{Q}$, $T\mathcal{C}$ can be embedded into $T\mathcal{Q}$ by the tangent lift $Ti : T\mathcal{C} \to T\mathcal{Q}$, and that admissible velocities are confined to the null space of the constraint Jacobian, $\boldsymbol{G}(\boldsymbol{q}) = D\boldsymbol{g}(\boldsymbol{q})$.

As in previous sections, admissible (non-interpenetration) configurations for $\boldsymbol{q}$ occupy the subset $\mathcal{A} \subset \mathcal{Q}$. To describe $\mathcal{A}$, we introduce a scalar valued constraint function for each pair of convex

*That is, if the exact indicator function is left in the Lagrangian, rather than being replaced by an approximate potential.

bodies in the system and index all pairings by the index set I_c

$$\mathcal{A} = \{\boldsymbol{q} \in \mathcal{Q} | g_i(\boldsymbol{q}) \leq 0 \forall i \in I_c\} \subset \mathcal{Q}. \tag{3.52}$$

Configurations for which $\boldsymbol{q}$ is in the boundary of $\mathcal{A}$, i.e $\boldsymbol{q} \in \partial\mathcal{A}$, are defined by $g_i(\boldsymbol{q}) = 0$.

For the constrained system, the equations of motion extremize an augmented Lagrangian of the form [53, 58]

$$\begin{aligned} \overline{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\lambda}) &= \dot{\boldsymbol{q}}^T \boldsymbol{M} \dot{\boldsymbol{q}} - V(\boldsymbol{q}) - \boldsymbol{g}^T(\boldsymbol{q})\boldsymbol{\lambda} \\ &= L(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}^T(\boldsymbol{q})\boldsymbol{\lambda}, \end{aligned} \tag{3.53}$$

where $\boldsymbol{M}$ is an appropriate inertial matrix, $V(\boldsymbol{q})$ is a holonomic potential. Taking variations of the action, $\int_0^T \overline{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\lambda})dt$, with respect to the state ($\boldsymbol{q}$, $\dot{\boldsymbol{q}}$, $\boldsymbol{\lambda}$) and the contact time t_c and enforcing stationarity gives

$$\begin{aligned} &\delta \int_0^T \overline{L}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{\lambda})dt \\ &\quad = \int_0^{t_c} \left(\frac{\partial L}{\partial \boldsymbol{q}} \cdot \delta\boldsymbol{q} + \frac{\partial L}{\partial \dot{\boldsymbol{q}}} \cdot \delta\dot{\boldsymbol{q}} - \boldsymbol{G}^T\boldsymbol{\lambda} \cdot \delta\boldsymbol{q} - \boldsymbol{g}^T\delta\boldsymbol{\lambda} \right) dt \\ &\qquad + \int_{t_c}^T \left(\frac{\partial L}{\partial \boldsymbol{q}} \cdot \delta\boldsymbol{q} + \frac{\partial L}{\partial \dot{\boldsymbol{q}}} \cdot \delta\dot{\boldsymbol{q}} - \boldsymbol{G}^T\boldsymbol{\lambda} \cdot \delta\boldsymbol{q} - \boldsymbol{g}^T\delta\boldsymbol{\lambda} \right) dt \\ &\qquad - \left[\!\left[\left(L(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}^T\boldsymbol{\lambda} \right) \delta t_c \right]\!\right]_{t_c^-}^{t_c^+} \\ &\quad = \int_0^T \left(\frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \boldsymbol{G}^T\boldsymbol{\lambda} \right) \delta\boldsymbol{q}dt - \int_0^T \left(\boldsymbol{g}^T\delta\boldsymbol{\lambda} \right) dt \\ &\qquad - \left[\!\left[\left(\frac{\partial L}{\partial \dot{\boldsymbol{q}}} \right) \cdot \delta\boldsymbol{q} + \left(L - \boldsymbol{g}^T\boldsymbol{\lambda} \right) \cdot \delta t_c \right]\!\right]_{t_c^-}^{t_c^+} \\ &\quad = 0, \end{aligned} \tag{3.54}$$

where in Equation (3.54), we have used $\delta\boldsymbol{q}(0) = \boldsymbol{0},\ \delta\boldsymbol{q}(T) = \boldsymbol{0}$, and integration by parts. The first two terms in (3.54) give the constrained equations of motion when contact is not taking place

$$\frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \boldsymbol{G}^T(\boldsymbol{q})\boldsymbol{\lambda} = \boldsymbol{0} \tag{3.55a}$$

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}, \tag{3.55b}$$

in which the term $-\boldsymbol{G}^T\boldsymbol{\lambda}$ represents internal constraint forces that prevent the system from deviating from the constraint manifold, $\mathcal{C}$. The final term in (3.54) gives the jump conditions at contact.

The constrained kinematic admissibility conditions are given by differentiating the relationship from the contact condition, $g(\boldsymbol{q}(t_c)) = 0$, accounting for the fact that $\boldsymbol{q} \in \mathcal{C}$ must also be met instantaneously in the contact configuration, i.e. $\boldsymbol{q}(t_c) \in \mathcal{A} \cap \mathcal{C} \equiv \mathcal{A}_\mathcal{C}$, with $\mathcal{A}_\mathcal{C}$ given by

$$\mathcal{A}_\mathcal{C} = \{\boldsymbol{q} \in \mathcal{C} | g_i(\boldsymbol{q}) \leq 0 \forall i \in I_c\} \subset \mathcal{C}. \tag{3.56}$$

In other words, $g(\boldsymbol{q}(t_c))$ must be differentiated on the constraint manifold $\mathcal{C}$, which can be accomplished by explicitly accounting for $\boldsymbol{q} \in \mathcal{C}$ via the map $\boldsymbol{f} : \mathcal{C} \to \mathcal{Q}$, which is related to the map used for the local reparametrization, $\boldsymbol{F}_d : \mathcal{U} \to \mathcal{Q}$, in that it reflects the fact that $\boldsymbol{q}$ is actually in a sub-manifold of $\mathcal{Q}$. As such, the contact constraint at the collision time t_c reads

$$g(\boldsymbol{f}(\boldsymbol{q}(t_c))) = 0,$$

which gives

$$\left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} \nabla g\right) \cdot (\delta \boldsymbol{q} + \dot{\boldsymbol{q}} \delta t_c) = 0, \; \nabla g \in \partial g, \text{ and} \tag{3.57}$$

where the abuse of notation is valid due to $\nabla g = \partial g$ if there is only one subgradient direction and, in the present development, a pre-selected direction $\nabla g \in \partial g$ based on geometric symmetries on the contact configuration is used in degenerate cases. The matrix $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}$ is a projection of $\nabla g \in T^*_{\boldsymbol{q}(t_c)}\mathcal{Q}$ onto $T^*_{\boldsymbol{q}(t_c)}\mathcal{C}^\dagger$, as illustrated for a system with a simple pendulum and a floor in Figure 3.34. Thus, the space of allowable variations for $(\delta \boldsymbol{q}, \delta t_c)$ are given by $\delta t_c = 0$ with $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} \nabla g \cdot \delta \boldsymbol{q} = \boldsymbol{0}$ for some $\nabla g \in \partial g$, and by $\delta \boldsymbol{q} = -\dot{\boldsymbol{q}} \delta t_c$. In essence, admissible variations, $\delta \boldsymbol{q}$, of $\boldsymbol{q}(t_c)$, are those in $T\partial\mathcal{A}_\mathcal{C}$ at the contact configuration. Substituting these variations and $\boldsymbol{g} = \boldsymbol{0}$ into the final term in (3.54) gives the impact equations

$$\left[\!\left[\frac{\partial L}{\partial \dot{\boldsymbol{q}}} \cdot \dot{\boldsymbol{q}} - L \right]\!\right]_{t_c^-}^{t_c^+} = 0, \tag{3.58a}$$

$$\left[\!\left[\frac{\partial L}{\partial \dot{\boldsymbol{q}}} \right]\!\right]_{t_c^-}^{t_c^+} \cdot \delta \boldsymbol{q} = 0, \; \forall \, \delta \boldsymbol{q} \text{ such that } \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} \nabla g \cdot \delta \boldsymbol{q} = \boldsymbol{0}, \; \nabla g \in \partial g, \tag{3.58b}$$

†Note that in the continuous constrained setting, the momenta $\boldsymbol{p}$ at configuration $\boldsymbol{q}$ neceessarily meet $\boldsymbol{p} \in T^*_{\boldsymbol{q}}\mathcal{C}$.

along with $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$. With this in hand, (3.58a) and (3.58b) reduce to

$$\left[\!\left[\boldsymbol{p}^T \boldsymbol{M}^{-1} \boldsymbol{p}\right]\!\right]_{t_c^-}^{t_c^+} = 0, \tag{3.59a}$$

$$\left[\!\left[\boldsymbol{p}\right]\!\right]_{t_c^-}^{t_c^+} = \mu_c \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}} \nabla g, \text{ for some } \nabla g \in \partial g, \tag{3.59b}$$

again with $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$. In (3.59b), $\mu_c \in \mathbb{R}$ is a Lagrange multiplier and $\boldsymbol{p} = \boldsymbol{M}\dot{\boldsymbol{q}}$. It will be convenient in future sections to define the full set of Lagrange multipiers associated with the contact configuration as $\boldsymbol{\mu}_c$, which is given by

$$\boldsymbol{\mu}_c = \begin{bmatrix} \boldsymbol{\lambda}_c \\ \mu_c \end{bmatrix}, \tag{3.60}$$

i.e., $\boldsymbol{\mu}_c$ is a vector with the first m entries equivalent to $\boldsymbol{\lambda}_c$, plus one additional entry, μ_c, for the additional constraint imposed by the contact configuration. Note that (3.59a) and (3.59b) represent jump conditions on the energy and momenta, stating that there is no change in the energy, and the the chance in momenta is orthogonal to the constraint Jacobian restricted to $T^*_{\boldsymbol{q}(t_c)}\mathcal{C}$.

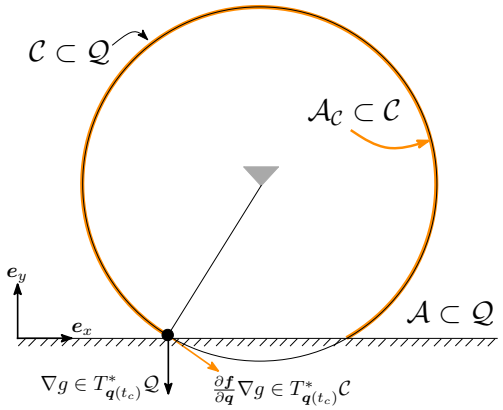


Figure 3.34: A constrained mechanical system composed of a simple pendulum with a contact constraint imposed by a floor. Here, $\boldsymbol{q} = (x, y)$ and the admissible set $\mathcal{A}$ is described by $g(\boldsymbol{q}) = -y$ being greater than or equal to zero. The sets $\mathcal{C}$, $\mathcal{A}$, and $\mathcal{A}_\mathcal{C}$ are illustrated, as well as the effect of the projection of ∇g by $\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{q}}$.

Null Space Approach. The continuous null space approach is discussed in detail in [50]. Here, the relevant results are summarized. The essence of the approach is that the constraint forces in (3.55) and (3.59b) can be eliminated by premultiplication with a null space matrix, $\boldsymbol{N}(\boldsymbol{q})$, which satisfies

$$\text{range}(\boldsymbol{N}(\boldsymbol{q})) = \text{null}(\boldsymbol{G}(\boldsymbol{q})) = T_{\boldsymbol{q}}\mathcal{C}, \tag{3.61}$$

and that the resulting equations of motion represent a D'Alembert type system since the premultiplication with a null space matrix is closely related to D'Alembert's principle that internal forces do no work [50].

3.4.2 Discrete Constrained Equations of Motion

This section describes the development of constrained discrete equations of motion for a trajectory with $K + \kappa$ time nodes and corresponding redundant configurations, $(\boldsymbol{q}_0, t_0), (\boldsymbol{q}_1, t_1), \ldots, (\boldsymbol{q}_{c_i^-}, t_{c_i^-})$, $(\boldsymbol{q}_{c_i}, t_{c_i}), (\boldsymbol{q}_{c_i^+}, t_{c_i^+}), \ldots, (\boldsymbol{q}_K, t_K)$, which combines the discrete conservation principles developed in [25, 50, 53], where the actual contact time(s), $t_{c_i} \in \left[t_{c_i^-}, t_{c_i^+}\right]$, and κ is the total number of collisions. The resulting discrete Euler-Lagrange equations lead directly to a variational integrator for the constrained mechanical system, which is identical to the integrator outlined in Section 3.2 away from the contact node, and modified in an analogous way to the modification of the continuous Euler-Lagrange equations near the contact node.

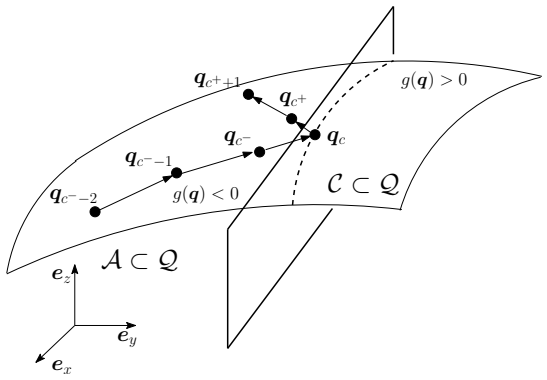


Figure 3.35: Evolution of the discrete configuration on the manifold $\mathcal{A} \subset \mathcal{Q}$.

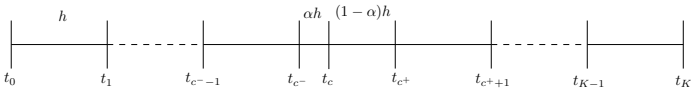


Figure 3.36: The discrete configurations are equally spaced in time away from the contact node, t_c.

To simplify the development in this section, only one collision event will be considered so that only $\kappa = 1$ intermediate configuration exists, as illustrated in Figures 3.35 and 3.36. For time constant time step h and total number of evenly spaced time nodes $K \in \mathbb{N}$ and also $c^-, c^+ \in \mathbb{N}$, let $\boldsymbol{q}_d : \{t_0, t_0 + h, \ldots, \ t_0 + (c^-)h, t_0 + \alpha h, t_0 + c^+ h \ldots, \ t_0 + Kh\} \to \mathcal{Q}$ be a discrete path in the configuration manifold $\mathcal{Q}$, and let $\boldsymbol{q}_k = \boldsymbol{q}_d(t_0 + kh) = \boldsymbol{q}_d(t_k)$. Take a discrete Lagrangian based

on a midpoint rule, $L_d : \mathcal{Q} \times \mathcal{Q} \to \mathbb{R}$ to approximate the continuous Lagrangian, $L : T\mathcal{Q} \to \mathbb{R}$, as follows

$$\begin{aligned} L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, h) &\approx \int_{t_k}^{t_{k+1}} L(\boldsymbol{q}, \dot{\boldsymbol{q}}) dt \\ L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, h) &= h \left[\frac{1}{2h^2} (\boldsymbol{q}_{k+1} - \boldsymbol{q}_k)^T \boldsymbol{M} (\boldsymbol{q}_{k+1} - \boldsymbol{q}_k) - V \left(\frac{(\boldsymbol{q}_{k+1} + \boldsymbol{q}_k)}{2} \right) \right]. \end{aligned} \tag{3.62}$$

To form the augmented discrete Lagrangian, the constraint term is discretized via

$$\frac{h}{2} \boldsymbol{g}^T(\boldsymbol{q}_k) \boldsymbol{\lambda}_k + \frac{h}{2} \boldsymbol{g}^T(\boldsymbol{q}_{k+1}) \boldsymbol{\lambda}_{k+1} \approx \int_{t_k}^{t_{k+1}} \boldsymbol{g}^T(\boldsymbol{q}) \boldsymbol{\lambda} dt. \tag{3.63}$$

$$\overline{L}_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{\lambda}_k, \boldsymbol{\lambda}_{k+1}, h) = L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, h) - \frac{h}{2} \boldsymbol{g}^T(\boldsymbol{q}_k) \boldsymbol{\lambda}_k - \frac{h}{2} \boldsymbol{g}^T(\boldsymbol{q}_{k+1}) \boldsymbol{\lambda}_{k+1}. \tag{3.64}$$

Note that in the preceding equations, an explicit dependence on the time step $h = t_{k+1} - t_k$ is allowed, in contrast to the basic development in Section 3.2. To derive the discrete Euler-Lagrange equations from an action sum, the time step is allowed to vary in the interval containing the actual contact time and configuration. That is, restricting $\alpha \in (0, 1)$, and the discrete action sum is given by

$$\begin{aligned} I_d = &\sum_{k=0}^{c^- - 1} \overline{L}_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{\lambda}_k, \boldsymbol{\lambda}_{k+1}, h) \\ &+ \overline{L}_d(\boldsymbol{q}_{c^-}, \boldsymbol{q}_c, \boldsymbol{\lambda}_{c^-}, \boldsymbol{\lambda}_c, \alpha h) + \overline{L}_d(\boldsymbol{q}_c, \boldsymbol{q}_{c^+}, \boldsymbol{\lambda}_c, \boldsymbol{\lambda}_{c^+}, (1 - \alpha) h) \\ &+ \sum_{k=c^+}^{K-1} \overline{L}_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{\lambda}_k, \boldsymbol{\lambda}_{k+1}, h). \end{aligned} \tag{3.65}$$

Taking variations of (3.65) with respect to $\boldsymbol{q}_k, \boldsymbol{\lambda}_k$, and α using $\delta \boldsymbol{q}_0 = \delta \boldsymbol{q}_K = \boldsymbol{0}$, and enforcing stationarity gives

$$
\begin{aligned}
\delta I_d = &\sum_{k=1}^{c^- -1} \left(D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k) + D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) - h\boldsymbol{G}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k \right) \cdot \delta \boldsymbol{q}_k \\
&+ \sum_{k=c^+ +1}^{K-1} \left(D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k) + D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) - h\boldsymbol{G}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k \right) \cdot \delta \boldsymbol{q}_k \\
&+ \left(D_2 L_d(\boldsymbol{q}_{c^- -1}, \boldsymbol{q}_{c^-}) + D_1 L_d(\boldsymbol{q}_{c^-}, \boldsymbol{q}_c, \alpha h) - \frac{(1+\alpha)h}{2} \boldsymbol{G}^T(\boldsymbol{q}_{c^-})\boldsymbol{\lambda}_{c^-} \right) \cdot \delta \boldsymbol{q}_{c^-} \\
&+ \left(D_2 L_d(\boldsymbol{q}_{c^-}, \boldsymbol{q}_c, \alpha h) + D_1 L_d(\boldsymbol{q}_c, \boldsymbol{q}_{c^+}, (1-\alpha)h) - \frac{h}{2} \boldsymbol{G}^T(\boldsymbol{q}_c)\boldsymbol{\lambda}_c \right) \cdot \delta \boldsymbol{q}_c \\
&+ \left(D_2 L_d(\boldsymbol{q}_c, \boldsymbol{q}_{c^+}, (1-\alpha)h) + D_1 L_d(\boldsymbol{q}_{c^+}, \boldsymbol{q}_{c^+ +1}) - \frac{(2-\alpha)h}{2} \boldsymbol{G}^T(\boldsymbol{q}_{c^+})\boldsymbol{\lambda}_{c^+} \right) \cdot \delta \boldsymbol{q}_{c^+} \\
&+ \left(D_3 L_d(\boldsymbol{q}_{c^-}, \boldsymbol{q}_c, \alpha h) - D_3 L_d(\boldsymbol{q}_c, \boldsymbol{q}_{c^+}, (1-\alpha)h) \right) h \delta \alpha \\
&+ \sum_{k=0}^{K-1} \boldsymbol{g}(\boldsymbol{q}_k) \cdot \delta \boldsymbol{\lambda}_k \\
&= 0,
\end{aligned}
\tag{3.66}
$$

where again, the notation $D_i L_d$ denotes the derivative of the discrete Lagrangian with respect to its i^{th} argument.

Pre- and post-collision. Away from the contact configuration, the discrete Euler-Lagrange equations for the augmented Lagrangian are given by

$$
D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, h) + D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k, h) - h\boldsymbol{G}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k = \boldsymbol{0}, \tag{3.67a}
$$

$$
\boldsymbol{g}(\boldsymbol{q}_{k+1}) = \boldsymbol{0}, \tag{3.67b}
$$

which can be solved as a system of equations for $\boldsymbol{q}_2, \ldots \boldsymbol{q}_{c^-}$ and $\boldsymbol{\lambda}_1, \ldots, \boldsymbol{\lambda}_{c^- -1}$ in the pre-collision configurations, and then $\boldsymbol{q}_{c^+ +1}, \ldots, \boldsymbol{q}_K$ and $\boldsymbol{\lambda}_{c^+}, \ldots, \boldsymbol{\lambda}_{K-1}$ in in the post collision configurations, assuming that the simulation is started so that no collision takes place in the first time step as described in Section 3.2.

Collision. In the interval surrounding the collision configuration, $\boldsymbol{q}_c$, a different set of equations must be solved. First, the equations are advanced according to (3.8) to $\tilde{\boldsymbol{q}}_{c^+}$ at t_{c^+}. If it is determined that $\tilde{\boldsymbol{q}}_{c^+} \notin \mathcal{A}$, then this inadmissible configuration is discarded, and Equations (3.68) are solved instead for α, $\boldsymbol{q}_c$, and $\boldsymbol{\lambda}_{c^-}$.

$$D_2L_d(\boldsymbol{q}_{c^--1},\boldsymbol{q}_{c^-})+D_1L_d(\boldsymbol{q}_{c^-},\boldsymbol{q}_c,\alpha h)-\frac{(1+\alpha)h}{2}\boldsymbol{G}^T(\boldsymbol{q}_{c^-})\boldsymbol{\lambda}_{c^-}=\boldsymbol{0} \tag{3.68a}$$

$$\boldsymbol{g}(\boldsymbol{q}_c)=\boldsymbol{0} \tag{3.68b}$$

$$g(\boldsymbol{q}_c)=0. \tag{3.68c}$$

With the collision configuration and time in hand, the simulation can the be advanced to $\boldsymbol{q}_{c^+}$, also solving for $\boldsymbol{\mu}_c$ according to

$$D_2L_d(\boldsymbol{q}_{c^-},\boldsymbol{q}_c,\alpha h)+D_1L_d(\boldsymbol{q}_c,\boldsymbol{q}_{c^+},(1-\alpha)h)-\frac{h}{2}\left[\boldsymbol{G}^T(\boldsymbol{q}_c)\,\nabla g(\boldsymbol{q}_c)\right]\boldsymbol{\mu}_c=\boldsymbol{0} \tag{3.69a}$$

$$\boldsymbol{g}(\boldsymbol{q}_{c^+})=\boldsymbol{0} \tag{3.69b}$$

$$D_3L_d(\boldsymbol{q}_{c^-},\boldsymbol{q}_c,\alpha h)-D_3L_d(\boldsymbol{q}_c,\boldsymbol{q}_{c^+},(1-\alpha)h)=0 \tag{3.69c}$$

where, in the analog to the continuous formulation, $\boldsymbol{\mu}_c=[\boldsymbol{\lambda}_c\,\mu_c]^T$. That is, an additional Lagrange multiplier, μ_c, has been introduced to enforce the constraint that variations of the collision configuration are restricted to the tangent space of the admissible set $\mathcal{A}$, i.e., that

$$\left(D_2L_d(\boldsymbol{q}_{c^-},\boldsymbol{q}_c,\alpha h)+D_1L_d(\boldsymbol{q}_c,\boldsymbol{q}_{c^+},(1-\alpha)h)-\frac{h}{2}\boldsymbol{G}^T(\boldsymbol{q}_c)\boldsymbol{\lambda}_c\right)\cdot\delta\boldsymbol{q}_c=0\ \forall\delta\boldsymbol{q}_c\in T\partial\mathcal{A}.$$

Note that Equation (3.69c) is a statement of conservation of *discrete* energy

$$\begin{aligned}E_d(\boldsymbol{q}_k,\boldsymbol{q}_{k+1},h)&=\frac{1}{2}\left(\frac{\boldsymbol{q}_{k+1}-\boldsymbol{q}_k}{h}\right)^T\boldsymbol{M}\left(\frac{\boldsymbol{q}_{k+1}-\boldsymbol{q}_k}{h}\right)+\boldsymbol{V}\left(\frac{\boldsymbol{q}_{k+1}+\boldsymbol{q}_k}{2}\right)\\&=KE_q+\boldsymbol{V}\left(\frac{\boldsymbol{q}_{k+1}+\boldsymbol{q}_k}{2}\right)\end{aligned}$$

so that (3.69c) can be written succinctly as

$$\begin{aligned}E_d(\boldsymbol{q}_{c^-},\boldsymbol{q}_c,\alpha h)-E_d(\boldsymbol{q}_c,\boldsymbol{q}_{c^+},(1-\alpha)h)&=0\\&=E_d^--E_d^+.\end{aligned} \tag{3.70}$$

Finally, care must also be taken that the appropriate time step is used to find the $\boldsymbol{q}_{c^+ +1}$ and $\boldsymbol{\lambda}_{c^+}$ according to

$$D_2 L_d(\boldsymbol{q}_c, \boldsymbol{q}_{c^+}, (1-\alpha)h) + D_1 L_d(\boldsymbol{q}_{c^+}, \boldsymbol{q}_{c^+ +1}, h) - \frac{(2-\alpha)h}{2} \boldsymbol{G}^T(\boldsymbol{q}_{c^+}) \boldsymbol{\lambda}_{c^+} = \boldsymbol{0}. \tag{3.71}$$

Hamiltonian Jump Conditions. In the continuous formulation, the energy jump condition (see Equation (3.59a)) was transformed via the continuous Legendre transform without mentioning to a jump condition on the Hamiltonian of the system. In the discrete setting, the Hamiltonian can be approximated by H_d defined as

$$\begin{aligned} H_d(\boldsymbol{q}_k, \boldsymbol{p}_k) &= \frac{1}{2} \boldsymbol{p}_k^T \boldsymbol{M}^{-1} \boldsymbol{p}_k + V(\boldsymbol{q}_k) \\ &= KE_p + V(\boldsymbol{q}_k). \end{aligned}$$

This energy approximates the continuous Hamiltonian $H(\boldsymbol{q}, \boldsymbol{p}) = \frac{1}{2} \boldsymbol{p}^T \boldsymbol{M}^{-1} \boldsymbol{p} + V(\boldsymbol{q})$, which is the energy of the continuous system. In fact, it is shown in [50] that in the absence of a potential, H_d exactly preserves the energy $H(\boldsymbol{q}_0, \boldsymbol{p}_0)$, in which the continuous state $(\boldsymbol{q}_0, \boldsymbol{p}_0)$ is used to start the discrete timestepping scheme, and that oscillations in H_d tend to be on a smaller magnitude than oscillations in E_d, even though both energies converge to H as the time step goes to 0.

Although it does not precisely conform to the discrete variational framework, the energy jump condition can be written on the momentum level using the projected Hamiltonian

$$\begin{aligned} {}^Q H_d(\boldsymbol{q}_k, \boldsymbol{p}_k) &= \frac{1}{2} \, {}^Q \boldsymbol{p}_k^T \boldsymbol{M}^{-1} \, {}^Q \boldsymbol{p}_k + V(\boldsymbol{q}_k) \\ &= {}^Q KE_p + V(\boldsymbol{q}_k) \end{aligned} \tag{3.72}$$

as

$$\begin{aligned} {}^Q H_d(\boldsymbol{q}_c, \boldsymbol{p}_c^+|_{\alpha h}) - {}^Q H_d(\boldsymbol{q}_c, \boldsymbol{p}_c^-|_{(1-\alpha)h}) &= 0 \\ &= {}^Q H_d^+ - {}^Q H_d^- \\ &= \left({}^Q KE_{p_c^+|_{\alpha h}} - {}^Q KE_{p_c^-|_{(1-\alpha)h}} \right) \end{aligned} \tag{3.73}$$

in which the notation $\boldsymbol{p}|_\gamma$ corresponds to slight modification of the discrete Legendre transform to be introduced shortly. This statement of energy conservation ammounts to conserving the part of $\boldsymbol{p}_k$ which lies in the same cotangent space as the continuous momentum. Furthermore, it corresponds to conserving the energy associated with the components of the discrete momenta avilable to do work in the system. It is shown in several examples that replacing the energy jump condition in terms of E_d with one in terms of ${}^Q H_d$ yields identical results in the pressence or absence of a potiential, and furthermore that computations using ${}^Q H_d$ are marginally faster than those using E_d.

It can be readily verified that (3.73) is equivalent to

$$\begin{aligned} {}^P H_d(\boldsymbol{q}_c, {}^P \boldsymbol{p}_c^+|_{\alpha h}) - {}^P H_d(\boldsymbol{q}_c, {}^P \boldsymbol{p}_c^-|_{(1-\alpha)h}) &= 0 \\ = {}^P H_d^+ - {}^P H_d^- & \\ = \left({}^P KE_{p_c^+|_{\alpha h}} - {}^P KE_{p_c^-|_{(1-\alpha)h}} \right) & \end{aligned} \tag{3.74}$$

in which the reduced mass matrix, $\boldsymbol{M}_{red} = \boldsymbol{P}^T(\boldsymbol{q}_c)\boldsymbol{M}\boldsymbol{P}(\boldsymbol{q}_c)$, should be used as the mass matrix in the final expression.

Discrete Null Space Approach and Reparametrization. The discrete constrained Euler-Lagrange equations (Equations 3.67, 3.68, and 3.69) can be reduced to a minimal set of equations by premultiplication of a discrete null space matrix and determining the update via a local nodal reparametrization in terms of the minimal set of configuration variables, $\boldsymbol{u} \in \mathcal{U}$, with $\mathcal{U}$ associated to $\mathbb{R}^{n-m}$. The resulting scheme is given by Equation (3.13) for pre-and post-collision intervals, and

$$\boldsymbol{P}^T(\boldsymbol{q}_{c^-}) \left[D_2 L_d(\boldsymbol{q}_{c^- -1}, \boldsymbol{q}_{c^-}) + D_1 L_d(\boldsymbol{q}_{c^-}, \boldsymbol{F}_d(\boldsymbol{u}_c, \boldsymbol{q}_{c^-}), \alpha h) \right] = \boldsymbol{0} \tag{3.75a}$$

$$g(\boldsymbol{F}_d(\boldsymbol{u}_c, \boldsymbol{q}_{c^-})) = 0. \tag{3.75b}$$

to be solved for $\boldsymbol{u}_c$ (and $\boldsymbol{q}_c$ via the parametrization) and α along with

$$\boldsymbol{P}^T(\boldsymbol{q}_c) \left[D_2 L_d(\boldsymbol{q}_{c^-}, \boldsymbol{q}_c, \alpha h) + D_1 L_d(\boldsymbol{q}_c, \boldsymbol{F}_d(\boldsymbol{u}_{c^+}, \boldsymbol{q}_c), (1-\alpha)h) - \frac{h}{2} \nabla g(\boldsymbol{q}_c) \mu_c \right] = \boldsymbol{0} \tag{3.76a}$$

$$E_d(\boldsymbol{q}_{c^-}, \boldsymbol{q}_c, \alpha h) - E_d(\boldsymbol{q}_c, \boldsymbol{q}_{c^+}, (1-\alpha)h) = 0, \tag{3.76b}$$

or the equivalent statement in terms of ${}^{Q}H_d$ to be solved for μ_c and $\boldsymbol{u}_c^+$ (and by extension $\boldsymbol{q}_c^+$) for the contact intervals.

Constrained Discrete Legendre Transform. The constrained discrete Legendre transforms defined in Equations 3.16 and 3.17, modified only slightly to account for the explicit dependence on the time step in the intervals near the contact node, can be used to formulate an alternative time-stepping scheme. Away from the contact configuration, we are left with

$$ {}^{P}\boldsymbol{p}_k^+ - {}^{P}\boldsymbol{p}_k^- = \boldsymbol{0}. $$

Near the contact configuration,

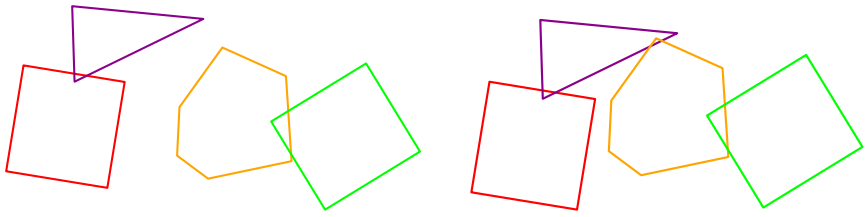
$$ \begin{aligned} {}^{P}\boldsymbol{p}_k^+ - {}^{P}\boldsymbol{p}_k^-|_{\alpha h} &= \boldsymbol{0} \\ g(\boldsymbol{q}_c) &= 0, \end{aligned} $$

and

$$ \begin{aligned} {}^{P}\boldsymbol{p}_k^+|_{\alpha h} - {}^{P}\boldsymbol{p}_k^-|_{(1-\alpha)h} &= \frac{h}{2}\,{}^{P}\nabla g\mu_c \\ {}^{P}H_d^- - {}^{P}H_d^+ &= 0. \end{aligned} $$

in which it is assumed that the parametrization $\boldsymbol{q}_k = \boldsymbol{F}_d(\boldsymbol{u}_k, \boldsymbol{q}_{k-1})$ is used, and the notation $\boldsymbol{p}|_\gamma$ implies that the momenta being considered is only associated with a portion of the interval h. Furthermore, the notation has been slightly abused on the right hand side of the expression so that ${}^{P}\nabla g\mu_c = \boldsymbol{P}^T(\boldsymbol{q}_c)\nabla g\mu_c$.

Multiple Collisions. If more than one pair of bodies are overlapping at the end of a non-contact step, two cases are considered. The first case, in which there are only pairwise collisions (see Figure 3.37) and no potentials which depend on the relative orientation of the bodies is easier to deal with. In this case, α is treated as a vector in Equation 3.75, with one entry for each independent collision. Having solved for $\boldsymbol{\alpha} \in \mathbb{R}^\kappa$, Equation 3.76 can be solved as-is, for $\mu_c \in \mathbb{R}$ and $\boldsymbol{q}_{c^+} = \boldsymbol{F}_d(\boldsymbol{u}_{c^+}, \boldsymbol{q}_c)$ with the vectored nature of $\boldsymbol{\alpha}$ taken into account.



(a) Case 1: pairwise collisions. (b) Case 2: *non*-pairwise collisions.

Figure 3.37: Two cases of multiple collisions. In the first case, only two bodies are involved in each collision event. In the second case, the order of collisions needs to be determined.

In the second case, as illustrated in Figure 3.37b, in which groups of overlapping bodies are detected or a potential is present which depends on the relative configuration of the bodies, a different method must be used because resolving the order of collisions becomes important to resolving the trajectory. Again, $\boldsymbol{\alpha}$ becomes a vector, but the equations are now coupled so that Equations 3.75 and 3.76 reflect the coupled nature of the collisions, i.e. terms like $(1-(\alpha_1+\alpha_2))$ appear, linking the update for the set of bodies involved in the multiple collision, and Equation 3.76 must be resolved in each sub-interval. Unfortunately, this formulation requires *a priori* knowledge of the oder of collisions, i.e., referring to Figure 3.38, one must know in advance *which* collision corresponds to the time $t_{c^-}+\alpha_1 h$, and which to $t_{c^-}+(\alpha_1+\alpha_2)h$ on order to accurately solve Equation 3.75.

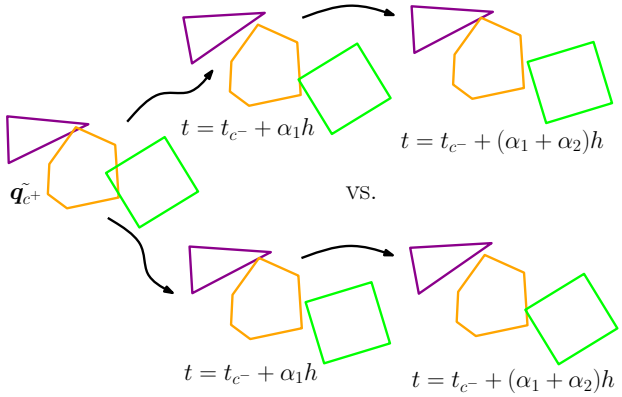


Figure 3.38: A priori knowledge of the order of collisions is needed to accurately solve Equation 3.75 for non-pairwise collisions.

A final complication comes from the fact that, in particular for the polygonal bodies under consideration, collisions cause relative rotation between the bodies, which in turn can lead to a series of collisions in quick succession between the same pair of bodies. That is, even for one pair-wise collision, advancing the configuration to $\boldsymbol{q}_{c^+}$ might *not* lead to an admissible configuration.

As a practical consideration to overcome the challenges associate with multiple collisions, if multiple collisions are detected, or if only pairwise collisions are detected but $\boldsymbol{q}_{c^+} \notin \mathcal{A}$, the time step h is repeatedly subdivided in the interval under consideration until there are only pairwise collisions

within each interval, and the configurations at the interval end points are admissible.

3.4.3 Examples

In the following examples, the constrained mechanical integrator developed in the preceding section is used in the $(\boldsymbol{q}, \boldsymbol{p})$ formulation and the reduced discrete Legendre transform, along with the appropriate parametrization are used to simulate systems involving elastic collisions between non-smooth bodies. The state of the system is advanced in the normal way until overlap is detected. Once detected, the overlapping configuration is thrown out and the discrete impact equations are solved to advance the simulation. Collisions between polyhedral bodies are detected using the SSH LP, which is also used to determine $\nabla g(\boldsymbol{q}_c)$. Unless otherwise stated, in all examples the energy shown is associated with the discrete Hamiltonian,

$$H_d(\boldsymbol{q}_k, \boldsymbol{p}_k) = \frac{1}{2}\boldsymbol{p}_k^T \boldsymbol{M}^{-1} \boldsymbol{p}_k + \boldsymbol{V}(\boldsymbol{q}_k),$$

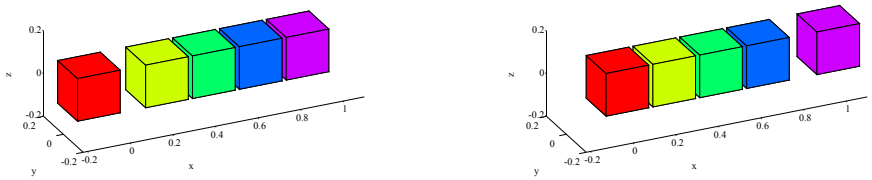
with the momenta $\boldsymbol{p}_k$ given by the constrained discrete Legendre transform, adjusted as needed to accommodate the variable time step in the contact intervals. The total angular momentum vector $\boldsymbol{L}$ for the discrete system can be recovered from the redundant momenta and configuration as

$$\boldsymbol{L} = \boldsymbol{\varphi} \times \boldsymbol{p}_\varphi + \boldsymbol{d}_I \times \boldsymbol{p}_I,$$

in which the repeated index I indicates summation. The labels 'Energy' and 'Angular Momentum' or $\boldsymbol{L}$ refer to these quantities unless otherwise noted. In the contact intervals, Matlab's built in nonlinear solver, fsolve, is used, with a trust region dog-leg method. The examples parallel the examples shown for the constrained mechanical integrator in Section 3.3, with the exception of the hexagon example, for reasons which will become obvious.

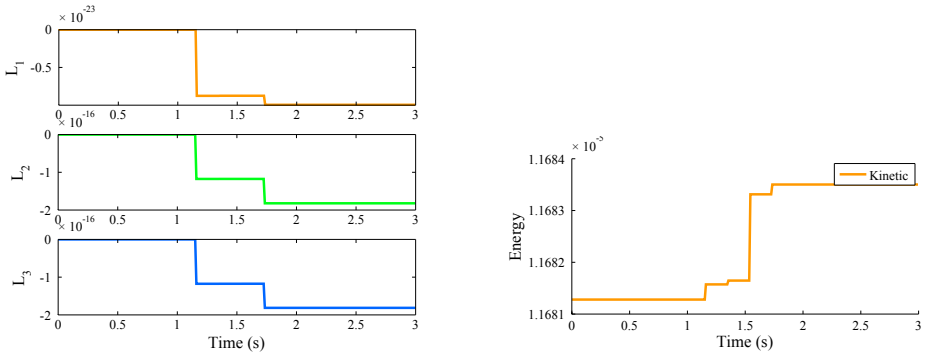
'Newton's Cradle'. As in the potential-based approach, two cases are considered. In case 1, four cubes are aligned at rest of side length $l = 0.2\sqrt{2}$m and uniform density $\rho = 270$ $\mathrm{kg/m^3}$ and separated by $l/10$. One cube of the same dimension starts with its face a distance of $l/2$ from the face of the end cube moves toward the bunch with initial velocity $\dot{\boldsymbol{\varphi}}^1 = [0.105\ 0\ 0]^T$ $\mathrm{m/s}$. As can be seen in the images in Figure 3.39, the impulse of the initial impact leads to the cube on the opposite end of the bunch from the original moving cube departing the cluster. The evolution of the energy and angular

momentum for this case are shown in Figure 3.40. The fixed time step is $h = 0.01$. In this example, only pairwise collisions take place, and the update to $\boldsymbol{q}_{c^+}$ is always an admissible configuration so that no special measures need to be taken to resolve the order of collisions.



(a) t = 0s (b) t = 3s

Figure 3.39: Initial and final configurations for first case of 'Newton's Cradle' examples.



(a) Angular Momentum. (b) Energy.

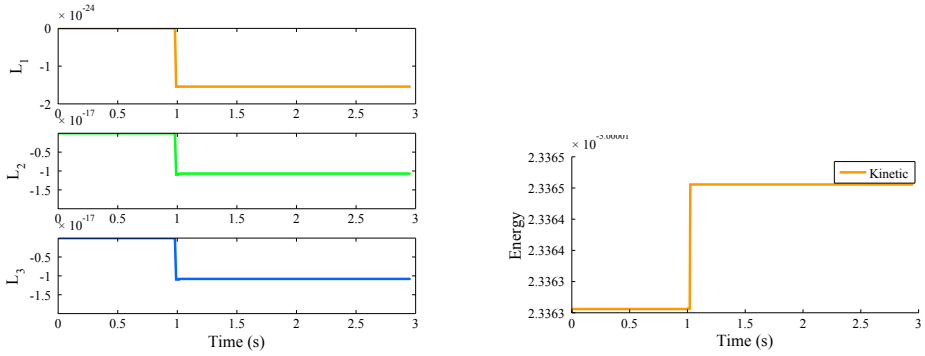
Figure 3.40: Angular momentum and energy evolution for and first case of 'Newton's Cradle' examples for $h = 0.01$s.

In case 2, three cubes are aligned at rest of side length $l = 0.2\sqrt{2}$ m and uniform density $\rho = 270$ $\mathrm{kg/m^3}$ and separated by $l/100$. The smaller spacing means that the time step must be adjusted to resolve the order of collisions. Two cubes of the same dimension start with faces a distance of $l/2$ from the face of the end cubes and move toward the bunch with initial velocities $\dot{\boldsymbol{\varphi}}^1 = [0.105\ 0\ 0]^T$ m/s and $\dot{\boldsymbol{\varphi}}^5 = [-0.105\ 0\ 0]^T$ m/s, respectively. As can be seen in the images in Figure 3.41, the impulse of the initial impact leads to both of the end cubes moving away from the cluster. The evolution of the energy and angular momentum for this case are shown in Figure 3.42. In this case, at least one non-pairwise multiple collision is unavoidable. Furthermore, for the given time-step, the initial velocity, and the proximity of the bodies, $\boldsymbol{q}_{c^+} \notin \mathcal{A}$, which means that the base time step must be decreased during a portion of the simulation, as shown in Figure 3.43.

In both of these examples, no relative rotation between the bodies is induced by the use of a the SSH LP to define $\nabla g(\boldsymbol{q}_c)$, which is the expected result for perfectly flat face-face contact. It should also be noted that for these examples, the results obtained from $E_d^+ = E_d^-$ and ${}^Q H_d^+ = {}^Q H_d^-$ are identical. However, the system with ${}^Q H_d^+ = {}^Q H_d^-$ is solved slightly faster, with a total CPU time of 10.962s

(a) t = 0s (b) t = 2.2s

Figure 3.41: Initial and final configurations for second case of 'Newton's Cradle' examples.



(a) Angular Momentum. (b) Energy.

Figure 3.42: Angular momentum and energy evolution for the second case of 'Newton's Cradle' examples for $h = 0.01$s.

versus 10.571s. This pattern is repeated–and more pronounced–in subsequent examples, suggesting the projected Hamiltonian constraint leads to a better conditioned set of equations.

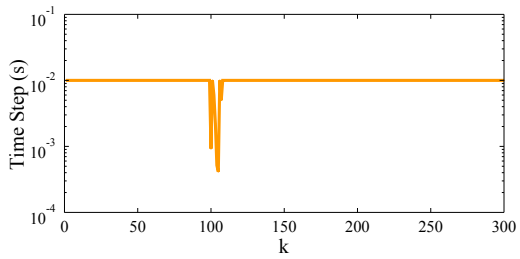
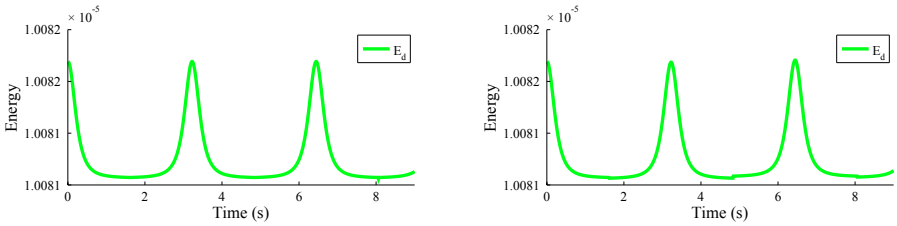


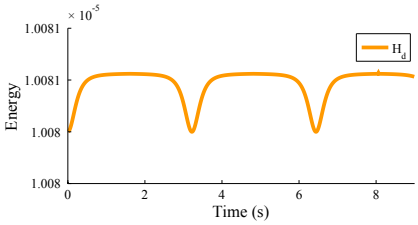
Figure 3.43: Time step used over the k^{th} interval for the second case the the 'Newton's Cradle' example. The smaller time steps reflect those needed to resolve the proper order for a series of collisions which take place in close succession.

Metronome. As in the analogous potential-based example, a simple kinematic chain consisting of two octohedral prisms with side length $0.2\sqrt{3}$ m connected by a spherical joint at the top of the first body and the bottom of the second body is introduced. In the initial configuration, $\boldsymbol{d}_3^1 = \boldsymbol{d}_3^2$, and the remaining directors of the second body are rotated by $\frac{\pi}{4}$rad about $\boldsymbol{d}_3$ relative to the first. The initial velocities used to start the simulation are $\boldsymbol{\omega}^1 = [0\ 1\ 0]^T$ rad⁄s and $\boldsymbol{\omega}^2 = [0\ -1\ 0]^T$ rad⁄s, so that the bodies rotate about the joint and swing towards each other, collide when the edge of the second body strikes the face of the first body, and so on. Several time lapse images of this motion are shown in Figure 3.45. As shown in Figure 3.44, oscillations in H_d are smaller in amplitude than oscillations in E_d

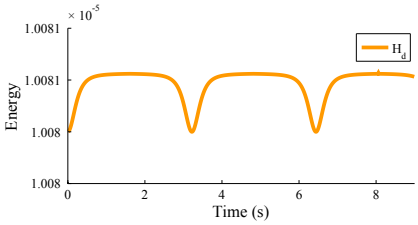


(a) Energy E_d using ${}^{Q}H_d^+ = {}^{Q}H_d^-$

(b) Energy E_d using $E_d^+ = E_d^-$



(c) Energy H_d using ${}^{Q}H_d^+ = {}^{Q}H_d^-$

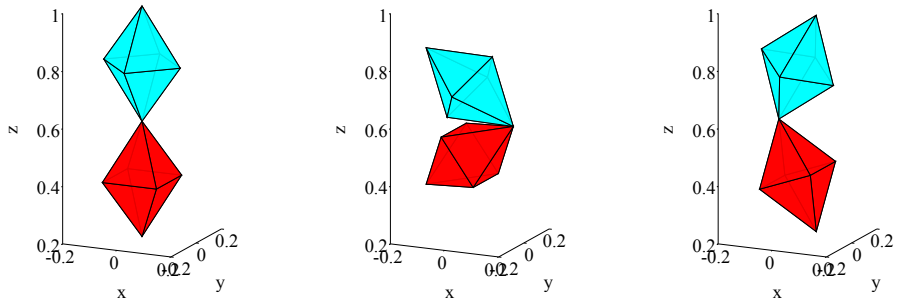


(d) Energy H_d using $E_d^+ = E_d^-$

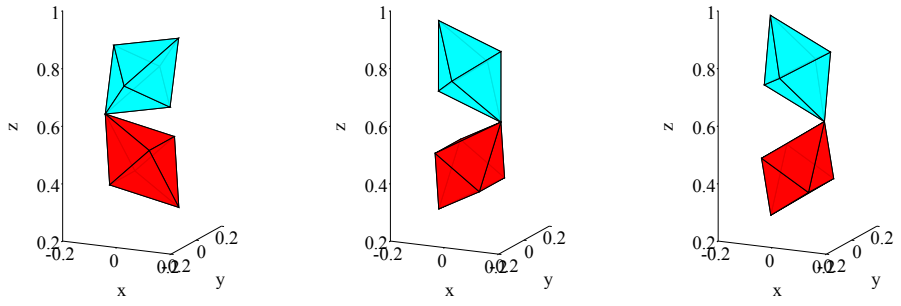
Figure 3.44: Energy in terms of E_d and H_d with the energy jump condifion in terms of either E_d or ${}^{Q}H_d$. H_d oscillates at a smaller amplitude than E_d, but the overall behavior of both schemes is indistinguishable.

The evolution of the energy and angular momentum for the metronome example using the discrete variational collision integrator and comparing the results using the discrete energy jump condition in terms of either E_d or ${}^{Q}H_d$ are shown in Figures 3.44 and 3.46, respectively. In the example shown, the fixed time step is $h = 0.015$s, and, clearly, only pairwise collisions take place. Furthermore, the simulation is initialized so that no additional relative rotation between the bodies is introduced, therefore precluding any need to modify h throughout the simulation the resolve to order of collisions.

As in the Newton's cradle examples, no additional relative rotation is induced by the use of the gradient from the SSH LP, which is the expected result for this particular edge-face contact configuration. The solution using the jump conditions in ${}^{Q}H_d$ as opposed to E_d again requires slightly less time to compute (6.15s versus 6.68s).

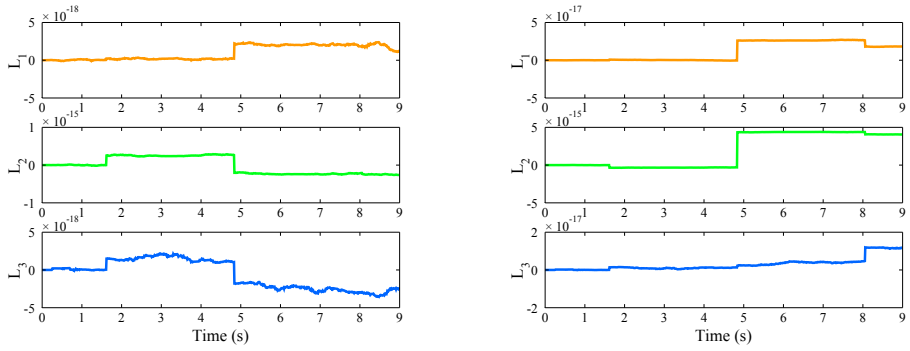


(a) $t = 0.0$s (b) $t = 1.8$s (c) $t = 3.6$s



(d) $t = 5.4$s (e) $t = 7.2$s (f) $t = 9.0$s

Figure 3.45: Time lapse images for a prismatic metronome connected by a spherical joint for the exact discrete variational integrator.



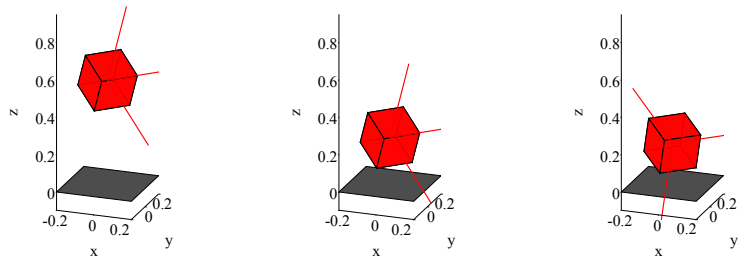
(a) Momentum, ${}^{Q}H_d^- = {}^{Q}H_d^+$. (b) Momentum, $E_d^- = E_d^+$.

Figure 3.46: Evolution of the total angular momentum $\boldsymbol{L}$ for the prismatic metronome solved with ${}^{Q}H_d^- = {}^{Q}H_d^+$ and $E_d^- = E_d^+$, respectively. The schemes are virtually indistinguishable and accurately conserve $\boldsymbol{L} = \boldsymbol{0}$.

Falling Polyhedra. Before moving to the larger system of seven polyhedra, the simple case of a cube colliding with a floor under the influence of a gravitational potential

$$V_g(\boldsymbol{q}) = -g\boldsymbol{e}_g^T \boldsymbol{M}\boldsymbol{q}$$

with $g = 9.81\text{m/s}^2$ and $\boldsymbol{e}_g^i = [\boldsymbol{e}_z\ \mathbf{0}\ \mathbf{0}\ \mathbf{0}]^T \in \mathbb{R}^{12}$ is investigated. The cube with properties described below is dropped with zero initial velocity from height of $\boldsymbol{\varphi} = 0.8\text{m}$. Time lapse images of the motion for $t \in [0.2, 0.8]\, s$ are shown in Figure 3.47 in order to confirm that the evolution of energy and momenta in the system is identical for ${}^{Q}H_d^- = {}^{Q}H_d^+$ and $E_d^- = E_d^+$, as shown in Figures 3.48 and 3.49. In all examples, the base time step is $h = 0.001\text{s}$.

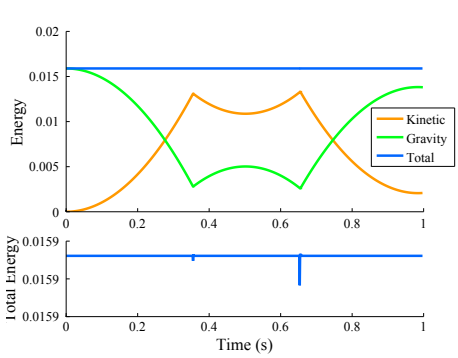


(a) $t = 0.2\text{s}$ (b) $t = 0.32\text{s}$ (c) $t = 0.44\text{s}$

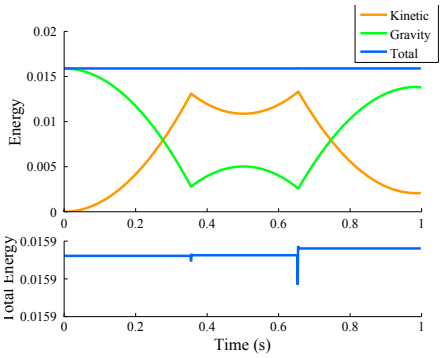
(d) $t = 0.56\text{s}$ (e) $t = 0.68\text{s}$ (f) $t = 0.8\text{s}$

Figure 3.47: Time lapse images for a falling cube under the influence of gravity.

In the analog to the example for the potential based approach, seven polyhedra start in a flower-like cluster with centers of mass $\sim 0.8\text{m}$ above a 'floor' occupying the $x - y$ plane, with a small $\sim 0.01\text{m/s}$ random initial translational velocity, and no initial angular velocity. For the purposes of this example, the floor is also modeled as a polyhedron, with vertices at $(x, y) = (\pm 10, \pm 10)\text{cm}$, and normal $\boldsymbol{e}_z$. In the evaluation of the SSH LP to determine overlap and ∇g for body-floor interactions, $\hat{\boldsymbol{\beta}} = \boldsymbol{e}_z$ is always used. Three different shapes are used; a cube with side length 0.2m, a right square pyramid with base dimension 0.2m and height 0.4m, and a regular octohedral prism with side length

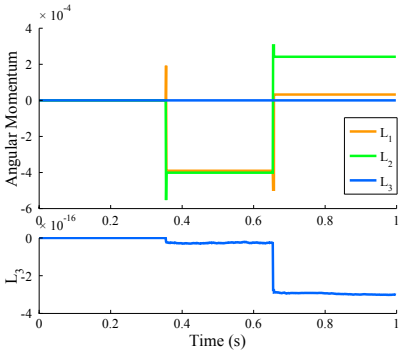


(a) Energy, ${}^{Q}H_d^- = {}^{Q}H_d^+$.

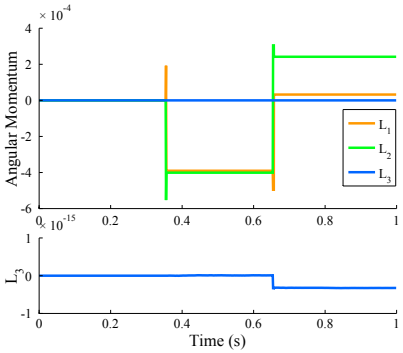


(b) Energy, $E_d^- = E_d^+$.

Figure 3.48: Conservation of H_d for discrete energy jump conditions ${}^{Q}H_d^- = {}^{Q}H_d^+$ and $E_d^- = E_d^+$.



(a) Momentum, ${}^{Q}H_d^- = {}^{Q}H_d^+$.

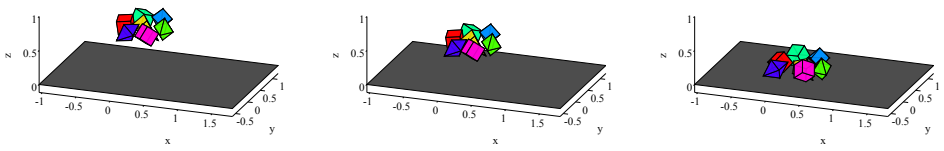


(b) Momentum, $E_d^- = E_d^+$.

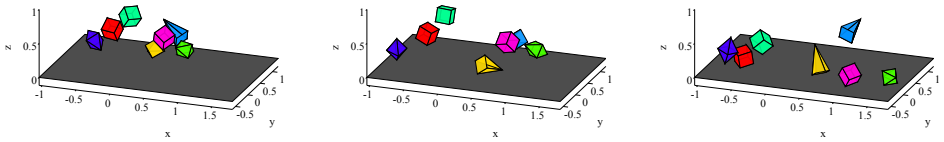
Figure 3.49: Conservation of L for discrete energy jump conditions ${}^{Q}H_d^- = {}^{Q}H_d^+$ and $E_d^- = E_d^+$ is essentially identical, with a slightly larger variation in L_3 for the second case.

$0.2\sqrt{3}$m, all with a uniform density of $\rho = 270^{\text{kg}/\text{m}^3}$. A series of time lapse images of the system is shown in Figure 3.50, and the evolution of the total energy and angular momentum is shown in Figure 3.51 for the E_d scheme and in in Figure 3.52 for the ${}^{Q}H_d$ scheme.

This example has several series of collisions occurring in close succession, including examples where induced rotation leads that same set of bodies (or a given body and the floor) to collide multiple times in different configurations, so that the time step must be adjusted accordingly, as shown in Figure 3.53. Again, both schemes are essentially equivalent.

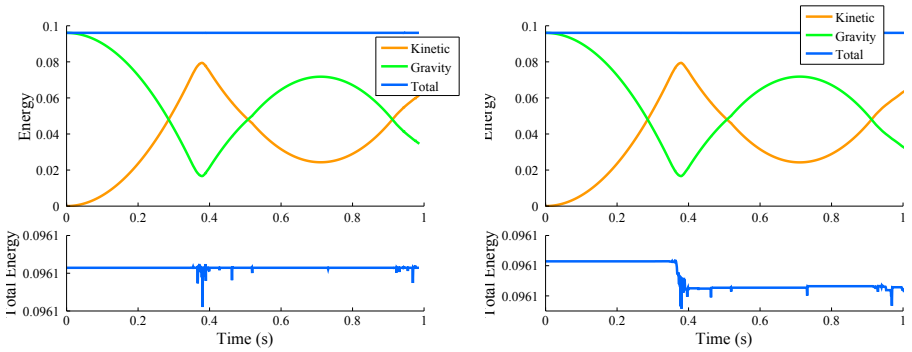


(a) $t = 0$s (b) $t = 0.2$s (c) $t = 0.4$s



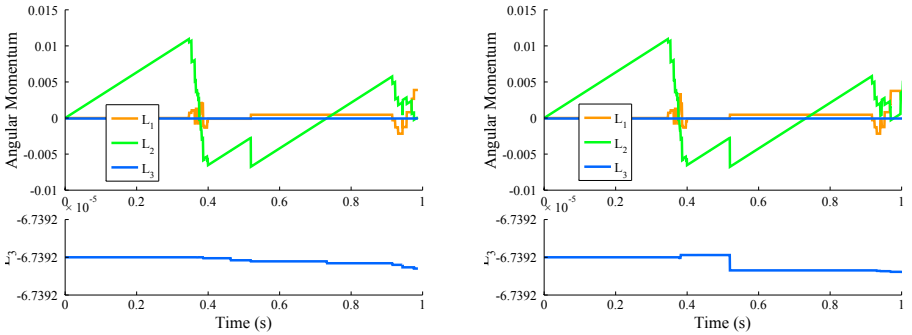
(d) $t = 0.6$s (e) $t = 0.8$s (f) $t = 1.0$s

Figure 3.50: Time lapse images for falling polyhedra using ${}^{Q}H_d^- = {}^{Q}H_d^+$ for discrete energy conservation.



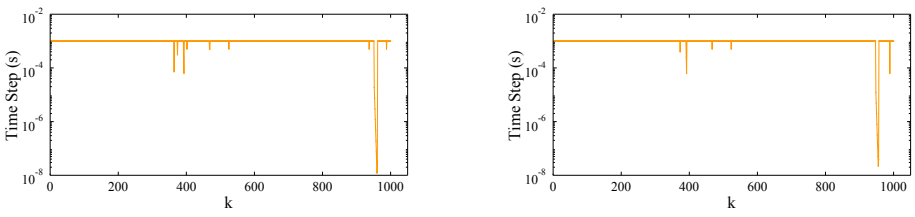
(a) Energy, ${}^{Q}H_d^- = {}^{Q}H_d^+$. (b) Energy, $E_d^- = E_d^+$.

Figure 3.51: Energy evolution for falling polyhedra with $h = 0.001$s comparing the results using ${}^{Q}H_d^- = {}^{Q}H_d^+$ and $E_d^- = E_d^+$.



(a) Angular Momentum, ${}^{Q}H_d^- = {}^{Q}H_d^+$. (b) Angular Momentum, $E_d^- = E_d^+$.

Figure 3.52: Angular momentum for falling polyhedra with $h = 0.001$s comparing the results using ${}^{Q}H_d^- = {}^{Q}H_d^+$ and $E_d^- = E_d^+$.



(a) Time step, $E_d^- = E_d^+$.

(b) Time step, ${}^Q H_d^- = {}^Q H_d^+$.

Figure 3.53: Time step for falling polyhedra with a base times step of $h = 0.001$s for $E_d^- = E_d^+$ and ${}^Q H_d^- = {}^Q H_d^+$ schemes, respectively.

Hexagonal Prisms. This example illustrates the limits of the discrete variational exact collision integration scheme in the presence of a potential which depends on the relative orientation of each body. In this case, the discrete impact equations cannot be decoupled for local collisions, i. e., the discrete impact equations must be solved the *full* system of bodies in order to advance the simulation, and not just for the subset of bodies involved in the collisions, which adds significantly to the expense of finding a solution. To see this consider that in the previous examples (including the polyhedra example), the vast majority of collisions which occur are pair-wise collisions[‡]. In this example, even *if* there are only pair-wise collisions, the problems are not local. This means that each contact interaction must be resolved separately–i. e. geometrically isolated pairwise interactions within the same step are not decoupled–and that the expense of solving the discrete impact equations in this example is due to solving the full $(n - m + 1)$-dimensional systems in the pre- and post- contact intervals (for $\boldsymbol{q}_c$ and α, and then $\boldsymbol{q}_{c^+}$ and μ_c, respectively), as opposed to just an $(n_{rb} - m_{rb} + 1)$-dimensional system. It goes without saying that this is further compounded by the fact that collisions between the bodies induce relative angular velocities, which in turn can lead to multiple collisions between the same pair of bodies within one time step.

As such, only results for a system of two hexagons are shown in detail. The hexagons have height 0.4m and a side length of 0.2m. As in previous examples, a series of alternating positive and negative Coulombic charges are centered behind each side face (see Figure 3.54), with the Coulombic potential for a single interaction given by (3.51), again with $\boldsymbol{r}_c^{j,i}$ given in terms of the directors for the j^{th}, $c_1, c_2 \in \{-1, 1\}$ are the sign of the charge, and K is a constant. In the present example, $K = 5e - 4$.

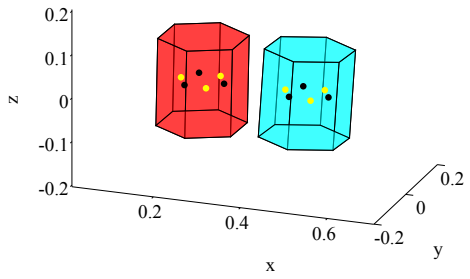
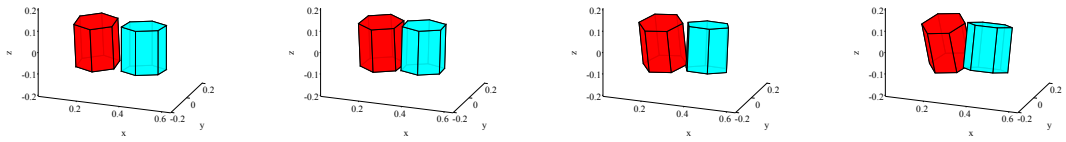


Figure 3.54: Hexagonal prisms shown with locations of alternating positive and negative (yellow and black) Coulombic charges.

Results for both schemes simulating the interaction of two hexaonal prisms (See time lapse images in Figure 3.55) were obtained, with the energy, angular momentum, and time step shown in Figures 3.56, 3.57, and 3.58, respectively for a base time step of $h = 0.01$s. Again, the ${}^{Q}H_d$ scheme results

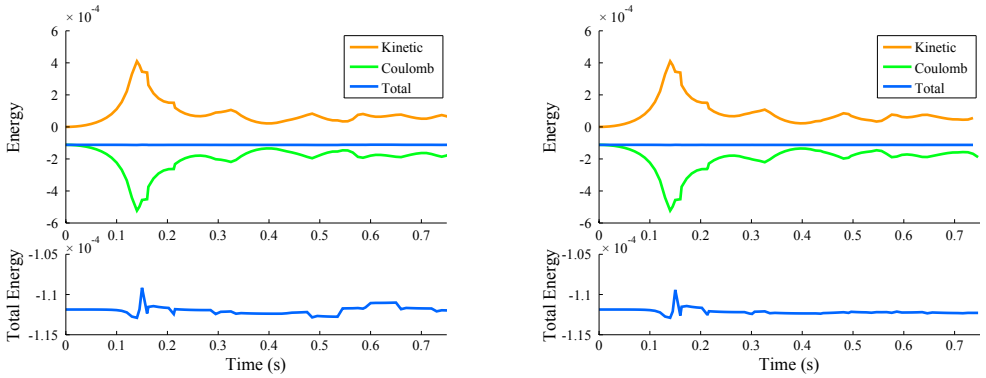
[‡]In fact, the only exception to this is the second Newton's cradle example in which a simultaneous collision occurs between three of the bodies.

in marginally smaller variations in the total energy and angular momentum. In addition, the total CPU time to run the E_d scheme for 80 intervals was 50.430s, compared to only 44.329s for the ${}^{Q}H_d$ scheme.



(a) $t = 0$s (b) $t = 0.19$s (c) $t = 0.38$s (d) $t = 0.74$s

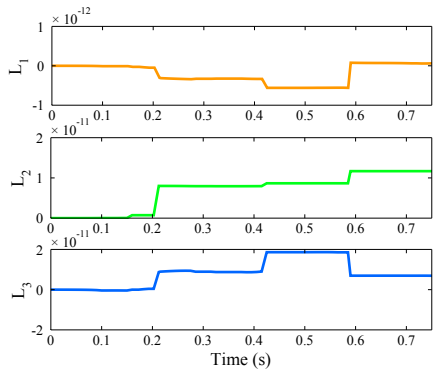
Figure 3.55: Time lapse images for two rigid hexagonal prisms with the motion driven by internal Coulombic charges, as shown in Figure 3.54.



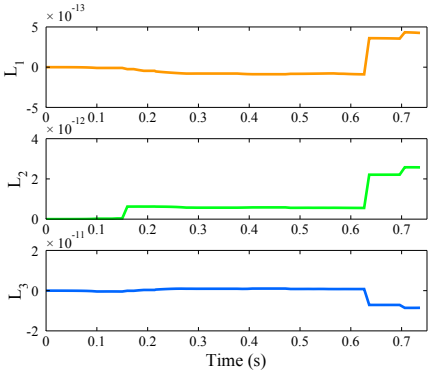
(a) Energy, $E_d^- = E_d^+$. (b) Energy, ${}^{Q}H_d^- = {}^{Q}H_d^+$.

Figure 3.56: Energy evolution for colliding hexagonal prisms with $h = 0.01$s comparing the results using ${}^{Q}H_d^- = {}^{Q}H_d^+$ and $E_d^- = E_d^+$.

An attempt was also made to run the full 7-body hexagonal prism example. Using the same parameters as in the 2-body hexagonal prism example, the CPU time to run for 40 time steps is $\sim$4500s. As shown in Figure 3.59, this number of steps only covers (using a base step of $h = 0.1$s only covers a simulation time of 0.3085s due to the need to switch to a smaller time step to resolve the order of multiple collisions.

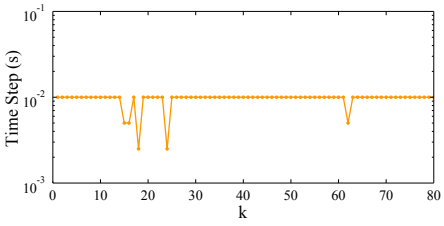


(a) Angular Momentum, $E_d^- = E_d^+$.

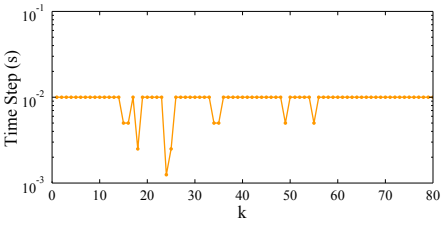


(b) Angular Momentum, ${}^Q H_d^- = {}^Q H_d^+$.

Figure 3.57: Angular momentum evolution for colliding hexagonal prisms with $h = 0.01$s comparing the results using ${}^Q H_d^- = {}^Q H_d^+$ and $E_d^- = E_d^+$.



(a) Time step, $E_d^- = E_d^+$.



(b) Time step, ${}^Q H_d^- = {}^Q H_d^+$.

Figure 3.58: Time step for system of two hexagonal prisms

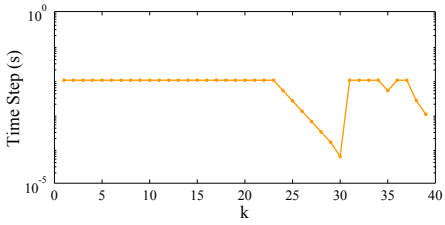


Figure 3.59: Time step for system of seven hexagonal prisms. The marked decrease in time step is to resolve the order of multiple collisions.

3.4.4 Discussion

In this section, a conservative discrete variational integrator has been introduced for constrained mechanical systems that closely follows the work of [25, 50, 53], and which is exact in the sense that the resulting trajectory exactly satisfies the discrete impact equations at the collision time(s). The essence of the scheme is to update the system in the normal way away from contact configurations. If overlap is detected, then the inadmissible configuration is thrown out, and additional variables $\boldsymbol{\alpha}$ are introduced to resolve the exact collision time. With the exact collision time(s) in hand, and in the analog to the continuous constrained equations of motion, the update near the contact configuration, $\boldsymbol{q}_c$, is completed by solving a set of jump conditions on the discrete conjugate momenta and energy. The latter condition can be stated either as an equivalence between the discrete energy, E_d, in the pre- and post-contact intervals, or the discrete restricted Hamiltonian, ${}^{Q}H_d$. However, in several examples using the projected $(\boldsymbol{q}, \boldsymbol{p})$ time-stepping scheme, the latter constraint seems to be the more natural constraint for the system in the sense that it lends itself to faster solution times marginally improved accuracy in the solution.

However accurate, this exact approach to resolving contact dynamics has distinct shortcomings. In particular, the exact order of multiple non-pairwise collisions during each time step needs to be resolved, and the post-collision configuration $\boldsymbol{q}_{c^+}$ must be admissible. Systems of polygonal bodies in which collisions induce relative rotation between the bodies are particularly problematic because the first collision in an interval can induce a series of successive collisions. Resolving the exact collision order can be accomplished by adaptively decreasing h when more complicated collisions are detected. However, the necessary decrease in time step can lead to inaccuracy or in the worst case, e.g. when potentials in the system drive the bodies towards contact configurations render the integrator prohibitively expensive.

With an eye on overcoming these shortcomings, Section 3.5 introduces a decomposition-based approach which avoids the need to exactly resolve the collision time.

3.5 Decomposition-Based Contact Dynamics

In Section 3.4, a constrained variational integrator which exactly conserves discrete momenta and energy through elastic collisions was developed using two different expressions of the energy for the discrete system. This integrator essentially relies on solving exactly for the times, $t_{c,i} \in \left[t_{c,i^-}, t_{c,i^+}\right]$, of the i^{th} collision between bodies (via solving for the variables α_i for each collision), and then solving discrete versions of the continuous energy and momentum jump conditions, which involve using a Lagrange multiplier to enforce $\delta \boldsymbol{q}_c \in T\mathcal{C} \cap T\partial\mathcal{A}$ at the collision in the momentum condition. In certain examples, the exact scheme can lose accuracy or completely stall when there are many bodies in the system, leading to many collisions per time step. Furthermore, in the formulation of optimal control problems, which is of interest in Chapter 4, this can lead to an *unknown* number of variables in the specification of the problem because in general one cannot expect to have a priori knowledge of the time or order for multiple collisions*.

This section develops the use of a so-called conserving approximation† in the form of a decomposition approach, along the lines of [15], which can be thought of as a limiting case of the exact approach introduced in the previous section. In particular, away from any collision events, the integration scheme is the constrained discrete variational integrator described in the previous section, so that appropriate discrete quantities are necessarily conserved according to Noether's theorem and the nature of the integrator. The key approximation is that all contact events take place *at the same time* at the time node following the interval in which the actual collisions take place, with the contact configuration approximated by a closest point projection operation which is enabled for polyhedral bodies by the use of the SSH LP for collision detection. As such, any non-physical changes to the conserved quantities in the system are exactly due to the projection operation, and can be tracked accordingly. Another advantage of this approach is that it enables the explicit inclusion either frictional forces or restitution within the momentum update, and precludes– at least for a simple approximation–the need to solve any additional conditions to determine these internal forces. This is in contrast to exact approaches which include frictional forces by solving a linear complementarity problem at the collision configuration, e.g. Leine et al. [46] and Pfeiffer and Glocker [73].

An additional advantage of this approach is that it can be extended in a straightforward manner to optimal control problems (to be discussed in Chapter 4), with a minimal introduction of new variables to the system. The approximate integration scheme described in this section extends the

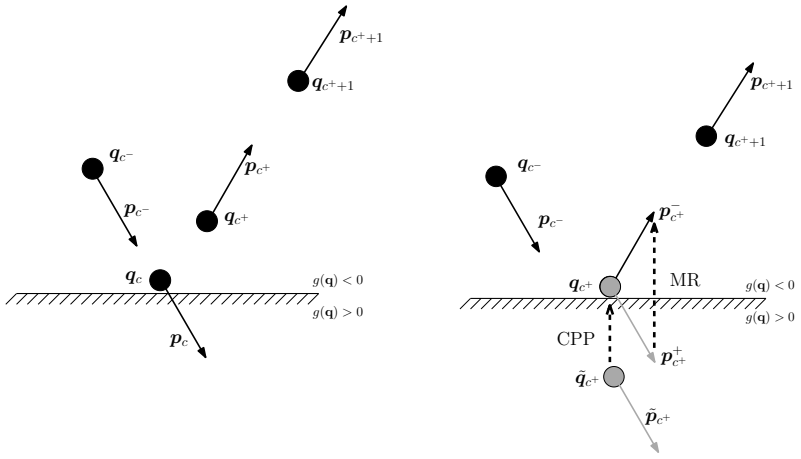
*Clearly, one cannot have a priori knowledge of whether or not collisions even take place. In the following section this will be overcome by assuming and pre-designating a number of collisions.

†In the decomposition-based approach, neither the discrete energy nor momenta will be conserved if a potential is present in the system. However, theses quantities chance only in a known way, and my a minimal amount. If there is no potential, then the discrete energy will be conserved.

Decomposition Contact Response (DCR) algorithm developed by [15] to constrained rigid body dynamics, and will henceforth be referred to as the DCR-Discrete Mechanics Constrained (DCR-DMC) algorithm.

3.5.1 Algorithm Overview.

To reiterate, the assumptions made in DCR-DMC are that all collisions in one time step take place simultaneously at the and of a time step, and that the collision configuration is given by a closest-point projection operation. This means that the collision response is decomposed into a closest point projection operation and the use of a momentum decomposition via a series of projection operations to update state of the system (see Figure 3.60).



(a) Exact solution [25]. (b) Explicit approximation [15].

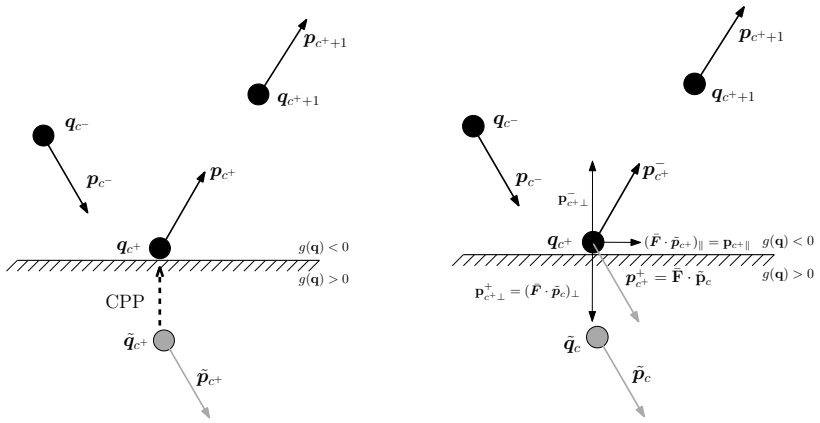
Figure 3.60: Exact solution resolving the collision time and configuration and the decomposition-based approximation to the impact equations (3.59a) and (3.59b). 'CPP' indicates that $\boldsymbol{q}_{c^+}$ is related to the inadmissible configuration $\tilde{\boldsymbol{q}}_c$ by a closest point projection operation, and 'MR' stands for the explicit *momentum reflection* operation which relates the pre-collision momenta $\boldsymbol{p}^+_{c^+}$ to the post-collision momenta $\boldsymbol{p}^-_{c^+}$

Algorithm 2 Decomposition Contact Response for Unconstrained Translational Dynamics [15]

1: $(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{p}})_k \leftarrow (\boldsymbol{q}, \boldsymbol{p})_{k-1}$
2: **if** $g(\tilde{\boldsymbol{q}}_k) > 0$ **then**
3: Let $\tilde{\boldsymbol{q}}_{k^+} = \tilde{\boldsymbol{q}}_k$ and $\boldsymbol{p}^+_{k^+} = \tilde{\boldsymbol{p}}_k$
4: $\boldsymbol{q}_{k^+} \leftarrow \tilde{\boldsymbol{q}}_{k^+}$ by closest-point projection
5: $\boldsymbol{p}^-_{k^+} \leftarrow \boldsymbol{p}^+_{k^+}$ by momentum decomposition
6: **else**
7: $(\boldsymbol{q}, \boldsymbol{p})_k \leftarrow (\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{p}})_k$
8: **end if**

A conceptual diagram of Algorithm 2 for a point-mass is shown in Figure 3.61. Note here that, in the presence of a potential or external forces are acting on the system, the total energy is not

conserved in the CPP operation. That is, if the system was driven to an inadmissible configuration potentials or external forcing, the CPP operation essentially 'un-does' some of the work done by these forces, hence changing the energy of the system.



(a) Closest point projection. (b) Momentum lift and reflection.

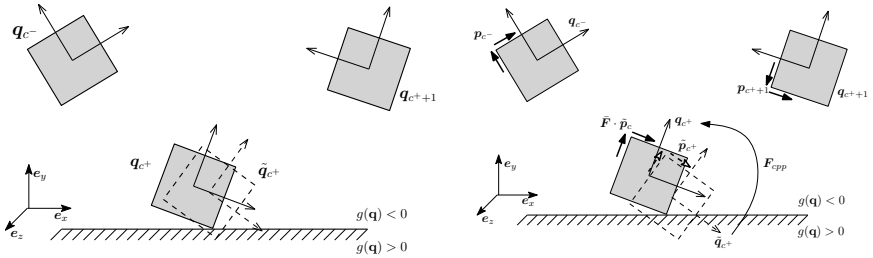
Figure 3.61: Conceptual diagram for Algorithm 2 for a point mass. $\bar{\boldsymbol{F}}$ represents the cotangent lift of $\tilde{\boldsymbol{p}}_c$ into the new subspace given by closest point projection, i.e. $\bar{\boldsymbol{F}} : T^*_{\tilde{q}_c} Q \mapsto T^*_{q_c} Q$. $\bar{\boldsymbol{F}}$ is the identity for the point mass system shown.

The DCR-DMC algorithm for rigid body dynamics 3 is quite similar to Algorithm 2 except for the inclusion of rotational degrees of freedom and the need to account for constraints. This necessitates the explicit inclusion of the mapping $\bar{\boldsymbol{F}} : T^*_{\tilde{q}_c} Q \mapsto T^*_{q_c} Q$, in the momentum reflection. This term is omitted from Algorithm 2 only because is it equal to the identity when rotational degrees of freedom are not present. This concept is illustrated in Figure 3.62.

Algorithm 3 Decomposition Contact Response–Discrete Mechanics Constrained

$(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{p}})_k \leftarrow (\boldsymbol{q}, \boldsymbol{p})_{k-1}$
if $g(\tilde{\boldsymbol{q}}_k) > 0$ **then**
  Let $\tilde{\boldsymbol{q}}_{k^+} = \tilde{\boldsymbol{q}}_k$
  $\boldsymbol{q}_{k^+} \leftarrow \tilde{\boldsymbol{q}}_{k^+}$ by closest-point projection
  Lift $\boldsymbol{p}^+_{k^+} = \bar{\boldsymbol{F}} \cdot \tilde{\boldsymbol{p}}_k$
  $\boldsymbol{p}^-_{k^+} \leftarrow \boldsymbol{p}^+_{k^+}$ by momentum decomposition
else
  $(\boldsymbol{q}, \boldsymbol{p})_k \leftarrow (\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{p}})_k$
end if

In order to understand the nature of the mapping $\bar{\boldsymbol{F}}$ and design consistent closest point projection and momentum reflection operations for rigid body dynamics, the formulation of the impact equations must first be revisited with the DCR-DMC approximations in mind.



(a) Closest point projection (b) Momentum lift and reflection

Figure 3.62: Conceptual diagram for Algorithm 3 for a square in two dimensions. $\bar{\boldsymbol{F}}$ represents the cotangent lift of $\tilde{\boldsymbol{p}}^+_{c^+}$ into the new subspace given by closest point projection. The redundant momentum $\boldsymbol{p}_I$ about the $z-$axis is shown in bold off-center arrows on the body. For simplicity, $\boldsymbol{p}_\varphi$ is not shown and the post-collision momenta are also not shown at configuration $\boldsymbol{q}_{c^+}$, but the effect of the collision is evident at configuration $\boldsymbol{q}_{c^+ +1}$.

3.5.2 Discrete Constrained Equations of Motion Revisited

In order to formulate the discrete equations of motion with the approximation that any contact events occur at a time node, a slightly different version of 3.66 is adopted, which accounts for the choice to never precisely resolve the exact collision times. For K equispaced time nodes, the discrete trajectory $(\boldsymbol{q}_0, t_0), (\boldsymbol{q}_1, t_1), \ldots, (\boldsymbol{q}_{c^- -1}, t_{c^- -1}), (\boldsymbol{q}_{c^-}, t_{c^-}), (\boldsymbol{q}_{c^+}, t_{c^+}), (\boldsymbol{q}_{c^+ +1}, t_{c^+ +1}), \ldots, (\boldsymbol{q}_K, t_K)$ is considered (see Figure 3.64).

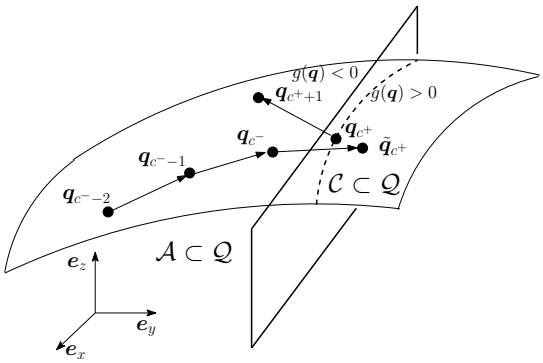


Figure 3.63: Evolution of the discrete configuration in the DCR-DMC scheme.

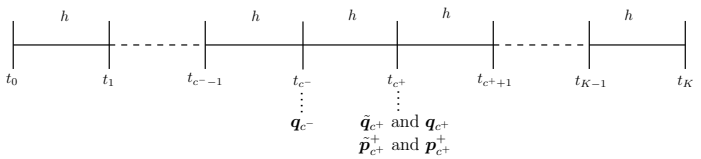


Figure 3.64: All time nodes are equispaced. However, *two* configurations are now associated with t_{c^+}, and *two* momenta are associated with the interval $[t_{c^-}, t_{c^+}]$, which are related by the closest point projection operation, and the cotangent lift induced on the momentum by this operation.

Working directly in the projected or reduced $(\boldsymbol{q}, \boldsymbol{p})$ scheme, and assuming that a local reparametriza-

tion so that the constraint vector $\boldsymbol{g}$ is eliminated from the system in the reduced scheme, the discrete Euler-Lagrange equations away from any contact nodes are given by

$$\begin{aligned} {}^{Q}\boldsymbol{p}_k^+ - {}^{Q}\boldsymbol{p}_k^- &= \mathbf{0}, \\ \boldsymbol{g}(\boldsymbol{q}_{k+1}) &= \mathbf{0} \end{aligned} \tag{3.77}$$

and

$${}^{P}\boldsymbol{p}_k^+ - {}^{P}\boldsymbol{p}_k^- = \mathbf{0}, \tag{3.78}$$

respectively.

The inadmissible configuration satisfies the constrained discrete (projected or reduced) Euler-Lagrange equations with constant time step h given by

$$\begin{aligned} {}^{Q}\boldsymbol{p}_{c^-}^+ - {}^{Q}\tilde{\boldsymbol{p}}_{c^-}^- &= \mathbf{0}. \\ \boldsymbol{g}(\tilde{\boldsymbol{q}}_c) &= \mathbf{0} \end{aligned}$$

and

$${}^{P}\boldsymbol{p}_{c^-}^+ - {}^{P}\tilde{\boldsymbol{p}}_{c^-}^- = \mathbf{0}.$$

in which $\tilde{\boldsymbol{p}}_{c^-}^- = -D_1 L_d(\boldsymbol{q}_{c^-}, \tilde{\boldsymbol{q}}_{c^+}) + \frac{h}{2}\boldsymbol{G}^T(\boldsymbol{q}_{c^-})\boldsymbol{\lambda}_{c^-}$, using the constrained discrete Legendre transform.

The unlifted momenta, ${}^{Q}\tilde{\boldsymbol{p}}_{c^+}$ and ${}^{P}\tilde{\boldsymbol{p}}_c$, are found with the discrete Legendre transforms

$${}^{\tilde{Q}}\tilde{\boldsymbol{p}}_c^+ = \boldsymbol{Q}(\tilde{\boldsymbol{q}}_c)\left[D_2 L_d(\boldsymbol{q}_{c^-}, \tilde{\boldsymbol{q}}_{c^+})\right]$$

and

$$ {}^{\tilde{P}}\tilde{\boldsymbol{p}}_c^+ = \boldsymbol{P}^T(\tilde{\boldsymbol{q}}_c)\left[D_2 L_d(\boldsymbol{q}_{c^-}, \tilde{\boldsymbol{q}}_{c^+})\right]. $$

Note that the use of the projected and reduced schemes precludes the need to solve for Lagrange multipliers associated with the inadmissible configuration $\tilde{\boldsymbol{q}}_c$. Once mapped to the proper subspace, the Lagrange multipliers associated with $\boldsymbol{q}_{c^+}$ can be recovered in post-processing, and used in the evaluation of the total energy and angular momentum of the system.

Using the function F_{cpp} to capture the closest point projection operation, $\boldsymbol{q}_{c^+} = F_{cpp}(\tilde{\boldsymbol{q}}_{c^+})$ along with ${}^{Q}\boldsymbol{p}_{c^+}^+ = \boldsymbol{Q}\bar{\boldsymbol{F}}_Q \cdot \tilde{\boldsymbol{p}}_{c^+}$[‡] and ${}^{P}\boldsymbol{p}_{c^+}^+ = \bar{\boldsymbol{F}}_P \cdot {}^{\tilde{P}}\tilde{\boldsymbol{p}}_{c^+}$. Noting that $g(\boldsymbol{q}_{c^+}) = 0$ and variations $\delta\boldsymbol{q}_{c^+}$ must lie in $\delta\boldsymbol{q}_{c^+} \in T\partial\mathcal{A}$, and making use of the Hamiltonian form of the energy jump conditions introduced in Section 3.4, the equations of motion through collisions for the DCR-DMC approximation are given by

$$ {}^{Q}KE_{\tilde{\boldsymbol{p}}_{c^+}} - {}^{Q}KE_{\boldsymbol{p}_{c^+}} = 0, \tag{3.79a} $$

$$ {}^{Q}\boldsymbol{p}_{c^+}^+ - {}^{Q}\boldsymbol{p}_{c^+}^- = h\mu_{c^+}\boldsymbol{Q}(\boldsymbol{q}_{c^+})\nabla g(\boldsymbol{q}_{c^+}), \tag{3.79b} $$

$$ \boldsymbol{g}(\boldsymbol{q}_{c^+ +1}) = \boldsymbol{0}, \tag{3.79c} $$

and

$$ {}^{P}KE_{\tilde{\boldsymbol{p}}_{c^+}} - {}^{P}KE_{\boldsymbol{p}_{c^+}} = 0, \tag{3.80a} $$

$$ {}^{P}\boldsymbol{p}_{c^+}^+ - {}^{P}\boldsymbol{p}_{c^+}^- = h\mu_{c^+}\boldsymbol{P}^T(\boldsymbol{q}_{c^+})\nabla g(\boldsymbol{q}_{c^+}). \tag{3.80b} $$

which are the *approximate* jump conditions to be solved for $\boldsymbol{q}_{c^+ +1}$ (or the local variables $\boldsymbol{u}_{c^+ +1}$) and $\boldsymbol{\mu}_c$, and it is assumed that the local reparametrization is use in (3.80). Note that the conservation of the kinetic part of the energy is due to the fact that if a potential is present in the system, it is unlikely that $V(\tilde{\boldsymbol{q}}_{c^+}) = V(\boldsymbol{q}_{c^+})$. However, if no potentials are present, then the approximate jump conditions exactly preserve the overall energy of the system. The end result of this formulation is that any changes to conserver quantities (i.e. $\boldsymbol{L}$ or H_d) are limited to those induced by the CPP operation to be discussed in Section 3.5.3.

Through the use of appropriate projections and decompositions, consistent explicit momentum up-

[‡]Note the slight abuse of notation in that $\tilde{\boldsymbol{p}}_{c^+}$ does not contain the constraint Jacobian term.

dates, or reflections, can be formulated so that (3.79b) and (3.80b) are met without explicitly solving for μ_c. That is, in the DCR-DMC algorithm, (3.79) and (3.80) are never solved as a system, but rather by an explicit momentum update after the CPP operation is used to resolve $\boldsymbol{q}_{c^+} \in \operatorname{bd} \mathcal{A}$.

3.5.3 Projection Operations for Constrained Rigid Bodies

The determination of an admissible configuration, $\boldsymbol{q}_{c^+}$, from the inadmissible configuration, $\tilde{\boldsymbol{q}}_{c^+}$ via a closest point projection (CPP) operation is the key approximation of the DCR-DMC algorithm. As such, this section gives an overview of the formal definition of the operation in a non-smooth setting, and then motivates and outlines operations which satisfy the approximate energy jump conditions.

Non-Smooth Analysis and CPP. In the non-smooth setting, a closest point projection is the resolvent of $\partial I_{\mathcal{A}}$,

$$R_{\partial I_{\mathcal{A}}} = (I + \partial I_{\mathcal{A}})^{-1},$$

which is, in general, a set-valued mapping from. For points $\boldsymbol{y} \notin \mathcal{A}$, the resolvent is a mapping to points $\boldsymbol{x} = R_{\partial I_{\mathcal{A}}}(\boldsymbol{y}) \in \operatorname{bd} \mathcal{A}$ given by (c.f. [16])

$$R_{\partial I_{\mathcal{A}}}(\boldsymbol{y}) = P_{\mathcal{A}}(\boldsymbol{y}) = \{\boldsymbol{x} \in \operatorname{bd} \mathcal{A} | \boldsymbol{x} \in \arg\min \|\boldsymbol{x} - \boldsymbol{y}\|\}.$$

CPP for DCR-DMC. In the DCR-DMC algorithm, the use of closest point projection to resolve an admissible configuration is not precisely rooted in non-smooth analysis[§]. Nor does it derived directly from a variational structure. Rather, it is rooted in the practical consideration of resolving an admissible configuration as quickly as possible, and the secondary goal of reducing any changes to conserved quantities in the system due to the projection operation. Before proceeding, we recall that the magnitude of $\boldsymbol{p}^+_{c^+}$ is entirely determined by $\tilde{\boldsymbol{p}}^+_{c^+}$, and the kinetic energy component of the projected or reduced Hamiltonian is explicitly conserved. What remains to be minimized in the approximate energy jump condition is the change in the potential energy due to the CPP operation. If no potentials are present in the system, then the goal of the CPP operation is to minimize the

[§]Such a scheme could be developed in the continuous setting along the lines of [38].

change to the total angular momentum due to the CPP operation, which amounts to minimizing the change in the configuration due to the CPP operation.

For example, $\boldsymbol{q}_{c^+} = \boldsymbol{F}_{cpp}(\tilde{\boldsymbol{q}}_{c^+})$ can be stated as a nonlinearly constrained quadratic program in the form

$$\begin{aligned} \boldsymbol{F}_{cpp}(\tilde{\boldsymbol{q}}_{c^+}) = &\arg\min_{\boldsymbol{q}\in\mathcal{Q}} \frac{1}{2h^2} \left(\boldsymbol{q} - \tilde{\boldsymbol{q}}_{c^+}\right)^T \boldsymbol{M} \left(\boldsymbol{q} - \tilde{\boldsymbol{q}}_{c^+}\right) \\ &\text{Subject to:} \\ &\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0} \\ &g_i(\boldsymbol{q}) = 0, \forall i \in I_{c,active}, \end{aligned} \tag{3.81}$$

in which the index set $I_{c,active}$ accounts for all potentially active pairwise collisions in the collision group being considered (see Figure 3.65). Experience has shown that the quadratic nature of the objective function combined with the locally convex nature of $g_i(\boldsymbol{q})$ near the inadmissible configuration means that the final constraint can be stated as an inequality, i. e. , $g_i(\boldsymbol{q}) \leq 0, \forall i \in I_{c,active}$, with the result that $g_i(\boldsymbol{q}) = 0$ for the same subset of $I_{c,active}$ as the solution to the equality constrained problem. With this in mind, to reduce Equation 3.81 to its minimal form, it is convenient to replace the final set of constraints with the unilateral contact potential energy, $V_{\mathcal{A}}(\boldsymbol{q})$, which is defined and used in (3.39) in Section 3.3. The parameter C can be used to adjust the tolerance of the contact constraint fulfillment at the solution. Furthermore, it is natural to eliminate the first set of constraints by a reparametrization in terms of $\boldsymbol{u}_{cpp}$, giving, and then setting $\boldsymbol{q}_{c^+} = \boldsymbol{F}_{cpp}(\boldsymbol{u}_{cpp}, \tilde{\boldsymbol{q}}_{c^+})$ as follows

$$\begin{aligned} \boldsymbol{u}_{cpp} = &\arg\min_{\boldsymbol{u}\in\mathcal{U}} \frac{1}{2h^2} \left(\boldsymbol{F}_d(\boldsymbol{u}, \tilde{\boldsymbol{q}}_{c^+}) - \tilde{\boldsymbol{q}}_{c^+}\right)^T \boldsymbol{M} \left(\boldsymbol{F}_d(\boldsymbol{u}, \tilde{\boldsymbol{q}}_{c^+}) - \tilde{\boldsymbol{q}}_{c^+}\right) \\ &\text{Subject to:} \\ &V_{\mathcal{A}}(\boldsymbol{F}_d(\boldsymbol{u}, \tilde{\boldsymbol{q}}_c)) \leq 0, \end{aligned} \tag{3.82}$$

which is equivalent to (3.81) at the solution, and now has only one constraint.

In the presence of a potential, a trade off can be made between minimizing changes to the angular momentum and total energy, and the CPP operation can be stated as

$$\begin{aligned}\boldsymbol{u}_{cpp} = \arg\min_{\boldsymbol{u}\in\mathcal{U}} \frac{1}{2h^2} \left(\boldsymbol{F}_d(\boldsymbol{u}, \tilde{\boldsymbol{q}}_{c^+}) - \tilde{\boldsymbol{q}}_{c^+}\right)^T \boldsymbol{M} \left(\boldsymbol{F}_d(\boldsymbol{u}, \tilde{\boldsymbol{q}}_{c^+}) - \tilde{\boldsymbol{q}}_{c^+}\right) + \|V(\boldsymbol{F}_d(\boldsymbol{u}, \tilde{\boldsymbol{q}}_{c^+})) - V(\tilde{\boldsymbol{q}}_{c^+})\| \\ \text{Subject to:} \\ V_{\mathcal{A}}(\boldsymbol{F}_d(\boldsymbol{u}, \tilde{\boldsymbol{q}}_{c^+})) \leq 0.\end{aligned} \tag{3.83}$$

For Lipschitz potentials such that

$$|V(\boldsymbol{q}) - V(\tilde{\boldsymbol{q}_{c^+}})| \leq K_L \|\boldsymbol{q} - \tilde{\boldsymbol{q}}_{c^+}\|$$

with Lipschitz constant K_L, the convexity of the objective function renders $g_i(\boldsymbol{q}) = 0,\ \forall i \in I_{c,active}$. However, the goal of the projection operation is to resolve a reasonable admissible contact configuration, with its structure also being motivated by the secondary–and somewhat imprecise–goal of minimizing changes to conserved quantities in the system. So, depending on the system being simulated, it is not absolutely necessary to include the potential energy in the objective functional, particularly if the potential is generally well behaved and does not decrease or increase drastically in any direction from $\tilde{q}_{c^+}$.

Multiple Contacts. Multiple contact scenarios pose significantly less of a problem for the DCR-DMC algorithm than for its exact counterpart due to the simultaneous contact assumption. Furthermore, the presence of potentials that depend on the relative configuration of the system (as in the hexagon examples, c.f. Section 3.4.3) does not preclude the decomposition of the CPP problem into a series of local collision groups, so long as it can be determined a sufficient buffer zone exists between the groups as shown in Figure 3.65. That being said, depending on the desired accuracy of a given simulation with respect to the overlap of the bodies, even if the bodies in different groups intersect at $\boldsymbol{q}_{c^+}$, the algorithm can proceed and this intersection will be dealt with at $\boldsymbol{q}_{c^+ + 1}$ if it still poses a problem. In the examples in this section, a buffer distance d_b of $d_b = 3 \max_{i\in I_c} g_i(\tilde{\boldsymbol{q}}_{c^+})$, or three times the maximum detected overlap, is used.

3.5.4 Momentum Decompositions for Constrained Rigid Bodies

As noted by [15], momentum decompositions are a particularly easy way to solve the impact equations if a contact configuration is known. The essence of these decompositions are to explicitly separate the normal (and sliding, if friction is considered) components of the momentum at the col-

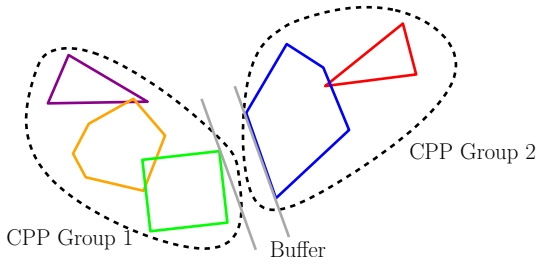


Figure 3.65: Decomposition of the CPP algorithm into two local problems with a buffer in between.

lision point and then modify these components according to the type of collision being considered. For example, for perfectly elastic collisions, the normal component of the momentum is reflected about the collision surface. If only translational degrees of freedom are present in the system, this method is especially simple. However, if angular degrees of freedom are present, then the redundant momenta are configuration dependent, and care must be taken to ensure that the momentum is decomposed and updated in the appropriate subspace.

Explicit momentum updates are sought which satisfy the requirements for the conserving approximation laid out in Section 3.5.1, i. e. , the proposed momentum updates do not contribute additional changes to conserved quantities *aside* from those due to the CPP operation. In the absence of a potential,

$$\begin{aligned} {}^{\tilde{Q}}H_d(\tilde{\boldsymbol{q}}_{c^+}, \tilde{\boldsymbol{p}}^+_{c^+}) &= {}^{Q}H_d(\boldsymbol{q}_{c^+}, \boldsymbol{p}^-_{c^+}), \qquad \text{or equivalently} \\ {}^{\tilde{P}}H_d(\tilde{\boldsymbol{q}}_{c^+}, {}^{\tilde{P}}\tilde{\boldsymbol{p}}^+_{c^+}) &= {}^{P}H_d(\boldsymbol{q}_{c^+}, {}^{P}\boldsymbol{p}^-_{c^+}) \end{aligned}$$

will be satisfied exactly. In the presence of a potential, $V(\tilde{\boldsymbol{q}}_c) = V(\boldsymbol{q}_c)$ may not result from the CPP operation. Thus, the goal of the algorithm is to always preserve the kinetic energy component of ${}^{Q}H_d$ or ${}^{P}H_d$

$$\begin{aligned} \frac{1}{2}\left(\tilde{\boldsymbol{p}}^+_{c^+}\right)^T \boldsymbol{M}^{-1}_{\tilde{Q}}\left(\tilde{\boldsymbol{p}}^+_{c^+}\right) &= \frac{1}{2}\left(\boldsymbol{p}^-_{c^+}\right)^T \boldsymbol{M}^{-1}_{Q}\left(\boldsymbol{p}^-_{c^+}\right), \qquad \text{or equivalently} \\ \frac{1}{2}\left({}^{\tilde{P}}\tilde{\boldsymbol{p}}^+_{c^+}\right)^T \tilde{\boldsymbol{M}}^{-1}_{red}\left({}^{\tilde{P}}\tilde{\boldsymbol{p}}^+_{c^+}\right) &= \frac{1}{2}\left({}^{P}\boldsymbol{p}^-_{c^+}\right)^T \boldsymbol{M}^{-1}_{red}\left({}^{P}\boldsymbol{p}^-_{c^+}\right). \end{aligned} \tag{3.84}$$

These equations should be solved in conjunction with

$$
\begin{aligned}
{}^{Q}\,\bar{\boldsymbol{F}}_{Q}\tilde{\boldsymbol{p}}^{+}_{c^+} - {}^{Q}\,\boldsymbol{p}^{-}_{c^+} &= h\mu_{c^+}\,{}^{Q}\,\nabla g(\boldsymbol{q}_{c^+}), \qquad \text{or} \\
\bar{\boldsymbol{F}}_{P}\,{}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^{+}_{c^+} - {}^{P}\,\boldsymbol{p}^{-}_{c^+} &= h\mu_{c^+}\,{}^{P}\,\nabla g(\boldsymbol{q}_{c^+}),
\end{aligned}
\tag{3.85}
$$

which is a restatement of (3.79b) and (3.80b) using the appropriate discrete constrained Legendre transforms and accounting for the need to lift the configuration-dependent redundant momenta to the new subspace defined by the post-CPP configuration. The mappings $\bar{\boldsymbol{F}}_Q$ and $\bar{\boldsymbol{F}}_P$ are defined in a natural way by the relationship between norms on the pre- and post-collision cotangent spaces.

Consistent Norms on $\eta(T^*\mathcal{C})$. In the projected formulation, the statement of energy conservation corresponds to

$$
\frac{1}{2}\left\langle {}^{\tilde{Q}}\,\tilde{\boldsymbol{p}}^{+}_{c^+}, {}^{\tilde{Q}}\,\tilde{\boldsymbol{p}}^{+}_{c^+}\right\rangle_{M^{-1}} = \frac{1}{2}\left\langle {}^{Q}\,\boldsymbol{p}^{-}_{c^+}, {}^{Q}\,\boldsymbol{p}^{-}_{c^+}\right\rangle_{M^{-1}}. \tag{3.86}
$$

All $(\tilde{\cdot})$ values are evaluated at the inadmissible configuration, and $(\cdot)$ values are either at the updated configuration or are configuration independent. The bracket notation $\langle \boldsymbol{x}, \boldsymbol{y}\rangle_{\boldsymbol{A}}$ denotes the $\boldsymbol{A}$-norm of $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e. the inner product $\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{y}$.

Equation (3.86) can be restated using this adjusted notation for the norm according to

$$
\frac{1}{2}\left\langle \tilde{\boldsymbol{p}}^{+}_{c^+}, \tilde{\boldsymbol{p}}^{+}_{c^+}\right\rangle_{M^{-1}_{\tilde{Q}}} = \frac{1}{2}\left\langle \boldsymbol{p}^{-}_{c^+}, \boldsymbol{p}^{-}_{c^+}\right\rangle_{M^{-1}_{Q}}, \tag{3.87}
$$

where $\tilde{\boldsymbol{p}}_{c^+}$ is the full redundant momentum associated with the inadmissible configuration, however due to the norm being on the projected subspace, it is not necessary to include the constraint Jacobian term. The necessary components of $\tilde{\boldsymbol{p}}_{c^+}$ are therefore readily available as the value inside of the brackets in equation $(3.16)_2$ or $(3.17)_2$ once $\tilde{\boldsymbol{q}}_{c^+}$ is known.

In (3.87), $\tilde{\boldsymbol{M}}^{-1}_{Q} = \tilde{\boldsymbol{Q}}^T\boldsymbol{M}^{-1}\tilde{\boldsymbol{Q}}$ and $\boldsymbol{M}^{-1}_{Q} = \boldsymbol{Q}^T\boldsymbol{M}^{-1}\boldsymbol{Q}$. Clearly, the terms on either side of equations (3.86) and (3.94) are norms in *different* subspaces. Therefore, in order to accomplish an explicit decomposition in the cotangent space of the *admissible* configuration, it is necessary to map the momenta in the inadmissible subspace into the admissible one. The necessary map can be recovered directly from the relationship between $\boldsymbol{q}_{c^+} = \boldsymbol{F}_{cpp}(\tilde{\boldsymbol{q}}_{c^+})$, and the relationship between the constraint Jacobian in each configuration. In particular, $\boldsymbol{G}(\boldsymbol{q})$ is related to $\boldsymbol{G}(\boldsymbol{q})$ by the relationship

$$\boldsymbol{G}(\boldsymbol{q}_{c^+}) = \tilde{\boldsymbol{G}}(\tilde{\boldsymbol{q}}_{c^+})\bar{\boldsymbol{F}}_Q^T. \tag{3.88}$$

And, for each body with $\boldsymbol{q} \in \mathbb{R}^{12}$,

$$\bar{\boldsymbol{F}}_Q = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{dR} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{dR} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{dR} \end{bmatrix} \in SO(12). \tag{3.89}$$

In (3.89), $\boldsymbol{I} = \boldsymbol{I}_{3\times 3}$, $\boldsymbol{0} = \boldsymbol{0}_{3\times 3}$, and $\boldsymbol{dR} \in SO(3)$ solves the equation

$$\boldsymbol{X} = \boldsymbol{dR}\tilde{\boldsymbol{X}}, \tag{3.90}$$

with $\boldsymbol{X} = [\boldsymbol{d}_1\ \boldsymbol{d}_2\ \boldsymbol{d}_3] \in SO(3)$.

That is, $\boldsymbol{dR}$ is the change in orientation made by the closest point projection operation. From the relationship

$$\tilde{\boldsymbol{G}}(\tilde{\boldsymbol{q}}_{c^+}) = \boldsymbol{G}(\boldsymbol{q}_{c^+})\bar{\boldsymbol{F}}_Q,$$

we find that

$$\begin{aligned} \tilde{\boldsymbol{Q}} &= \boldsymbol{I} - \bar{\boldsymbol{F}}_Q^T\boldsymbol{G}^T(\boldsymbol{G}\bar{\boldsymbol{F}}_Q\boldsymbol{M}^{-1}\bar{\boldsymbol{F}}_Q^T\boldsymbol{G}^T)^{-1}\boldsymbol{G}\bar{\boldsymbol{F}}_Q\boldsymbol{M}^{-1} && \text{(3.91a)} \\ &= \boldsymbol{I} - \bar{\boldsymbol{F}}_Q^T\boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{M}^{-1}\boldsymbol{G}^T)^{-1}\boldsymbol{G}\boldsymbol{M}^{-1}\bar{\boldsymbol{F}}_Q && \text{(3.91b)} \\ &= \bar{\boldsymbol{F}}_Q^T\left[\boldsymbol{I} - \boldsymbol{G}^T(\boldsymbol{G}\boldsymbol{M}^{-1}\boldsymbol{G}^T)^{-1}\boldsymbol{G}\boldsymbol{M}^{-1}\right]\bar{\boldsymbol{F}}_Q && \text{(3.91c)} \\ &= \bar{\boldsymbol{F}}_Q^T\boldsymbol{Q}\bar{\boldsymbol{F}}_Q && \text{(3.91d)} \end{aligned}$$

due to the inertial matrix, $\boldsymbol{M}$, being diagonal. Application of the results of (3.91) to the left hand side of (3.87) yields

$$\frac{1}{2}\left\langle \tilde{\boldsymbol{p}}_{c^+}^+, \tilde{\boldsymbol{p}}_{c^+}^+ \right\rangle_{\tilde{M}_Q^{-1}} = \frac{1}{2}\left\langle \bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}^+, \bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}^+ \right\rangle_{M_Q^{-1}}. \tag{3.92}$$

Therefore, we can ensure that equation (3.87) is satisfied, by requiring

$$\frac{1}{2}\left\langle \bar{\boldsymbol{F}}_Q \tilde{\boldsymbol{p}}^+_{c^+}, \bar{\boldsymbol{F}}_Q \tilde{\boldsymbol{p}}^+_{c^+} \right\rangle_{M_Q^{-1}} = \frac{1}{2}\left\langle \boldsymbol{p}^-_{c^+}, \boldsymbol{p}^-_{c^+} \right\rangle_{M_Q^{-1}}, \tag{3.93}$$

with the norms before and after reflection now being over the same subspace, with the mapping $\bar{\boldsymbol{F}}_Q : T^*_{\tilde{q}}\mathcal{Q} \mapsto T^*_q\mathcal{Q}$ defined in (3.89).

Consistent Norms on $T^*\mathcal{U}$. In the reduced formulation, the energy jump condition reads

$$\frac{1}{2}\left\langle {}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^+_{c^+}, {}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^+_{c^+} \right\rangle_{\tilde{M}_{red}^{-1}} = \frac{1}{2}\left\langle {}^{P}\,\boldsymbol{p}^-_{c^+}, {}^{P}\,\boldsymbol{p}^-_{c^+} \right\rangle_{M_{red}^{-1}}. \tag{3.94}$$

Which is equivalent to (3.86) due to the identity

$$\boldsymbol{Q}^T\boldsymbol{M}^{-1}\boldsymbol{Q} = \boldsymbol{P}\boldsymbol{M}_{red}^{-1}\boldsymbol{P}^T, \tag{3.95}$$

We first note that, for free rigid bodies, $\boldsymbol{M}_{red}$ may be written as

$$\boldsymbol{M}_{red} = \begin{bmatrix} \boldsymbol{I}m_\varphi & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}\boldsymbol{J}_0\boldsymbol{X}^T \end{bmatrix}, \tag{3.96}$$

where m_φ is the mass of the body associated with translational degrees of freedom, and $\boldsymbol{J}_0$ is the (principle) inertia tensor about the center of mass. This may be explicitly inverted as

$$\boldsymbol{M}_{red}^{-1} = \begin{bmatrix} \boldsymbol{I}m_\varphi^{-1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}\boldsymbol{J}_0^{-1}\boldsymbol{X}^T \end{bmatrix}. \tag{3.97}$$

Again, we must affect a map from $T^*_{\tilde{u}}\mathcal{U}$ to $T^*_u\mathcal{U}$ so that the momentum is consistently decomposed in an appropriate subspace. We can deduce this lift from the relationship

$$\boldsymbol{M}_{red}^{-1} = \begin{bmatrix} \boldsymbol{I}m_\varphi & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{X}\boldsymbol{J}_0^{-1}\boldsymbol{X}^T \end{bmatrix} = \bar{\boldsymbol{F}}_P \tilde{\boldsymbol{M}}_{red}^{-1} \bar{\boldsymbol{F}}_P^T \tag{3.98}$$

$$\bar{\boldsymbol{F}}_P = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{dR} \end{bmatrix}. \tag{3.99}$$

Applying $\bar{\boldsymbol{F}}_P$ to the left hand side of (3.94), the following equality holds

$$\frac{1}{2}\left\langle {}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^{+}_{c^+}, {}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^{+}_{c^+}\right\rangle_{\tilde{M}^{-1}_{red}} = \frac{1}{2}\left\langle \bar{\boldsymbol{F}}_P\ {}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^{+}_{c^+}, \bar{\boldsymbol{F}}_P\ {}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^{+}_{c^+}\right\rangle_{M^{-1}_{red}}. \tag{3.100}$$

Therefore, the energy jump condition in the reduced frame may be written as

$$\frac{1}{2}\left\langle \bar{\boldsymbol{F}}_P\ {}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^{+}_{c^+}, \bar{\boldsymbol{F}}_P\ {}^{\tilde{P}}\,\tilde{\boldsymbol{p}}^{+}_{c^+}\right\rangle_{M^{-1}_{red}} = \frac{1}{2}\left\langle {}^{P}\,\boldsymbol{p}^{-}_{c^+}, {}^{P}\,\boldsymbol{p}^{-}_{c^+}\right\rangle_{M^{-1}_{red}}. \tag{3.101}$$

Kinematic Chains. For kinematic chains, it is most convenient to conserve energy in the reduced norm. To accomplish this, we note that for a given null space matrix $\boldsymbol{P}^C$, the redundant momenta $\boldsymbol{p}^{+P^C}_{c^+}$ can be recovered from the reduced momenta as

$$\boldsymbol{p}^{+P^C}_{c^+} = \boldsymbol{M}\boldsymbol{P}^C\left(\boldsymbol{M}^C\right)^{-1}_{red}\ {}^{P^C}\,\boldsymbol{p}^{+}_{c^+},$$

a fact which is used to start the simulation, with $\boldsymbol{\nu}_0 = \left(\boldsymbol{M}^C\right)^{-1}_{red}\ {}^{P^C}\,\boldsymbol{p}_0$. According to the preceding developments in this section, the redundant momenta can be mapped from $T^*_{\tilde{\boldsymbol{q}}}\mathcal{Q}$ to $T^*_{\boldsymbol{q}}\mathcal{Q}$ by applying a rotation $\boldsymbol{dR}$ to the $\boldsymbol{p}_I$ components of the redundant momenta so that they are given in the post-CPP director frame. So, in order to work in the reduced subspace for kinematic chains, the mapping $\bar{F}_Q$ can *first* be applied to the redundant momenta (not including the constraint Jacobian terms). That is, for a general kinematic chain[¶], the energy jump condition in the reduced norm is given by

$$\frac{1}{2}\left\langle {}^{P^C}\,\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}^{+}_{c^+}, {}^{P^C}\,\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}^{+}_{c^+}\right\rangle_{(M^C)^{-1}_{red}} = \frac{1}{2}\left\langle {}^{P^C}\,\boldsymbol{p}^{-}_{c^+}, {}^{P^C}\,\boldsymbol{p}^{-}_{c^+}\right\rangle_{(M^C)^{-1}_{red}}, \tag{3.102}$$

in which ${}^{P^C}\,\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+} = \boldsymbol{P}^{C^T}\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}$.

3.5.4.1 Frictionless Contact.

With the redundant mapped to the post-CPP location and the corresponding appropriate statements of energy conservation in terms of the restricted Hamiltonian, the DCR-DMC algorithm proceeds by identifying the normal and tangential components of the momentum. Using the normality property of the gradient of the interpenetration constraint $\nabla g(\boldsymbol{q}_{c^+})$,

[¶]As noted in passing in the previous paragraph, this formulation could also be used for free rigid bodies, although it is less convenient.

$$\boldsymbol{p} = \boldsymbol{p}_{norm} + \boldsymbol{p}_{tang}, \tag{3.103}$$

at the contact point.

Let us first consider the constrained scheme. In terms of the induced norms on $\eta(T^*\mathcal{C})$, the tangential component of the momentum satisfies

$$\begin{aligned} \langle \nabla g, \boldsymbol{p}_{tang} \rangle_{M_Q^{-1}} = 0, \text{ or equivalently,} \\ \left\langle {}^{Q}\nabla g, {}^{Q}\boldsymbol{p}_{tang} \right\rangle_{M^{-1}} = 0, \end{aligned} \tag{3.104}$$

where the projection of ∇g by $\boldsymbol{Q}$, here denoted ${}^{Q}\nabla g$, can be interpreted as selecting the portion of the (redundant) gradient ${}^{Q}\nabla g \in \eta(T^*_{\boldsymbol{q}}\mathcal{C})$ capable of inducing changes in ${}^{Q}\boldsymbol{p} \in \eta(T^*_{\boldsymbol{q}}\mathcal{C})$. This is consistent with (3.85); only momentum components in the direction $\boldsymbol{Q}\nabla g$ should be altered by the collision. Equation (3.105) states that tangential momenta, ${}^{Q}\boldsymbol{p}_{tang}$ occupy a subspace orthogonal that defined by $\mathcal{Q}\nabla g$. Substituting (3.103) into (3.104) leads to

$$\left\langle {}^{Q}\nabla g, {}^{Q}\boldsymbol{p} - {}^{Q}\boldsymbol{p}_{norm} \right\rangle_{\boldsymbol{M}^{-1}} = 0, \tag{3.105}$$

so that the normal component of the momentum can be found explicitly via an orthogonal projection as

$${}^{Q}\boldsymbol{p}_{norm} = \frac{\langle \nabla g, \boldsymbol{p} \rangle_{\boldsymbol{M}_Q^{-1}}}{\langle \nabla g, \nabla g \rangle_{\boldsymbol{M}_Q^{-1}}} \, {}^{Q}\nabla g. \tag{3.106}$$

The reduced formulation yields the analogous result. That is, ${}^{P}\boldsymbol{p}_{tang}$ satisfies

$$\begin{aligned} \langle {}^{P}\nabla g, {}^{P}\boldsymbol{p}_{tang} \rangle_{\boldsymbol{M}_{red}^{-1}} = 0, \text{ or, with (3.103),} \\ \langle {}^{P}\nabla g, {}^{P}\boldsymbol{p} - {}^{P}\boldsymbol{p}_{norm} \rangle_{\boldsymbol{M}_{red}^{-1}} = 0. \end{aligned} \tag{3.107}$$

Now, the projection of the ∇g by $\boldsymbol{P}^T$ identifies the appropriate portion of the (redundant) gradient ${}^{P}\nabla g \in T^*_{\boldsymbol{u}}\mathcal{U}$ capable of inducing changes in ${}^{P}\boldsymbol{p} \in T^*_{\boldsymbol{u}}\mathcal{U}$. Equation (3.107) states that tangential momenta, ${}^{P}\boldsymbol{p}_{tang}$ occupy a subspace orthogonal that defined by the restriction of ∇g to $T^*_{\boldsymbol{u}}\mathcal{U}$.

Again, the normal component of the momentum can be found explicitly via an orthogonal projection as

$$ {}^{P}\boldsymbol{p}_{norm} = \frac{\langle {}^{P}\nabla g, {}^{P}\boldsymbol{p}\rangle_{\boldsymbol{M}_{red}^{-1}}}{\langle {}^{P}\nabla g, {}^{P}\nabla g\rangle_{\boldsymbol{M}_{red}^{-1}}} \, {}^{P}\nabla g. \tag{3.108} $$

In the absence of frictional forces, the tangential components of the momentum remain fixed, i.e. only the components of the momentum in the direction of ${}^{P}\nabla g$ or ${}^{Q}\nabla g$ will be affected by the collision. Inelastic frictionless collisions can be modeled using a coefficient of restitution $e \in [0,1]$, with $e = 1$ being perfectly elastic, and $e = 0$ perfectly plastic. Using the results of this and the preceding sections, explicit update for the post-collision momenta can be updated according to

$$ {}^{Q}\boldsymbol{p}_{c^+}^{-} = {}^{Q}\bar{\boldsymbol{F}}_{Q}\tilde{\boldsymbol{p}}_{tang,c^+}^{+} - e\,{}^{Q}\bar{\boldsymbol{F}}_{Q}\tilde{\boldsymbol{p}}_{norm,c^+}^{+} \tag{3.109a} $$

$$ {}^{P}\boldsymbol{p}_{c^+}^{-} = \bar{\boldsymbol{F}}_{P}\,{}^{P}\tilde{\boldsymbol{p}}_{tang,c^+}^{+} - e\bar{\boldsymbol{F}}_{P}\,{}^{P}\tilde{\boldsymbol{p}}_{norm,c^+}^{+}, \tag{3.109b} $$

from which it can be seen via (3.86,3.94) that the energies (3.84,3.85) are exactly preserved for $e = 1$. Furthermore, Equation (3.109) may be expressed in terms of the normal impulse $\boldsymbol{I}_{norm}(\boldsymbol{p}) = -(1+e)\boldsymbol{p}_{norm}$, giving

$$ {}^{Q}\boldsymbol{p}_{c^+}^{-} = {}^{Q}\bar{\boldsymbol{F}}_{Q}\tilde{\boldsymbol{p}}_{c^+}^{+} + \boldsymbol{I}_{norm}({}^{Q}\bar{\boldsymbol{F}}_{Q}\tilde{\boldsymbol{p}}_{c^+}^{+}) \tag{3.110a} $$

$$ {}^{P}\boldsymbol{p}_{c^+}^{-} = \bar{\boldsymbol{F}}_{P}\,{}^{\tilde{P}}\tilde{\boldsymbol{p}}_{c^+}^{+} + \boldsymbol{I}_{norm}(\bar{\boldsymbol{F}}_{P}\,{}^{\tilde{P}}\tilde{\boldsymbol{p}}_{c^+}^{+}). \tag{3.110b} $$

Finally, before moving on to formulations for frictional contact, note that the tangential component of the momentum ($\boldsymbol{p}_{tang}$) can be further decomposed into fixed $\boldsymbol{p}_{fix}$ and sliding $\boldsymbol{p}_{slide}$ components, in which the velocities associated with fixed components do not cause any relative motion between the bodies at the point of impact, and the sliding components represent motions along a contact surface, which will be related to frictional forces.

3.5.4.2 Frictional Contact.

This section develops an impulse based approach to model Coulombic frictional contact between two bodies. To begin, note that the momentum can be decomposed into subspaces which described the fixed and non-fixed components, i.e.

$$\boldsymbol{p} = \boldsymbol{p}_{fix} + \boldsymbol{p}_{nonfix}, \tag{3.111}$$

where, extending [15]'s definition, $\boldsymbol{p}_{fix}$ is the component of the momentum associated with projected or reduced velocities $\boldsymbol{M}_Q^{-1}\boldsymbol{p} \in T_{\boldsymbol{q}}\mathcal{C}$ or $\boldsymbol{M}_{red}^{-1}\,{}^P\,\boldsymbol{p} \in T_{\boldsymbol{u}}\mathcal{U}$ which instantaneously keeps the separation vector $\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{0}$. The separation vector $\boldsymbol{h}(\boldsymbol{q})$ is naturally defined by the KKT conditions of the SSH LP at the optimum by

$$\boldsymbol{h}(\boldsymbol{q}) = \boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2 + \boldsymbol{d}_I^1 \sum_{i \in I_1} v_{i,I}\pi_i - \boldsymbol{d}_I^2 \sum_{i \in I_2} v_{i,I}\pi_i.$$

Recall Section 3.3.2 for details. The nonfixed part of the projected momenta is composed of both the normal ($\boldsymbol{p}_{norm}$) and sliding ($\boldsymbol{p}_{slide}$) components of the momentum, and the fixed components represent a subset of the tangential ($\boldsymbol{p}_{tang}$) momentum representing rigid body motions under which the separation function, g, is invariant. To isolate the nonfixed momenta in the appropriate cotangent space, it is recognized that the fixed components of the momentum from the projected and reduced formulations, ${}^Q\,\boldsymbol{p}_{fix}$ and ${}^P\,\boldsymbol{p}_{fix}$ must satisfy

$$\boldsymbol{h}(\boldsymbol{q}(t_{c^+} + \epsilon)) = \boldsymbol{h}(\boldsymbol{q}(t_{c^+})) + \nabla\boldsymbol{h}\boldsymbol{M}_Q^{-1}\boldsymbol{p}_{fix}\epsilon + O(\epsilon^2) = \nabla\boldsymbol{h}\boldsymbol{Q}^T\boldsymbol{M}^{-1}\,{}^Q\,\boldsymbol{p}_{fix} = \boldsymbol{0}, \text{ and} \tag{3.112a}$$

$$\boldsymbol{h}(\boldsymbol{q}(t_{c^+} + \epsilon)) = \boldsymbol{h}(\boldsymbol{q}(t_{c^+})) + \nabla\boldsymbol{h}\boldsymbol{P}\boldsymbol{M}_{red}^{-1}\,{}^P\,\boldsymbol{p}_{fix}\epsilon + O(\epsilon^2) = \boldsymbol{0}, \tag{3.112b}$$

in which $\epsilon \ll 1$. To recapitulate, $\boldsymbol{p}_{fix}$ is associated with motions that instantaneously leave $\boldsymbol{h} = \boldsymbol{0}$, which implies that

$$\nabla\boldsymbol{h}\boldsymbol{Q}^T\boldsymbol{M}^{-1}\,{}^Q\,(\boldsymbol{p} - \boldsymbol{p}_{nonfix}) = \boldsymbol{0}, \text{ and} \tag{3.113a}$$

$$\nabla\boldsymbol{h}\boldsymbol{P}\boldsymbol{M}_{red}^{-1}\,{}^P\,(\boldsymbol{p} - \boldsymbol{p}_{nonfix}) = \boldsymbol{0}, \tag{3.113b}$$

or, that the fixed (projected and reduced) momentum components, ${}^Q\,\boldsymbol{p}_{fix}$ and ${}^P\,\boldsymbol{p}_{fix}$, span the null space of the columns gradient of the separation vector, projected into the relevant subspaces, ${}^Q\,\nabla\boldsymbol{h}^T$ and ${}^P\,\nabla\boldsymbol{h}^T$, with respect to the inner product induced by $\boldsymbol{M}^{-1}$ or $\boldsymbol{M}_{red}^{-1}$, respectively. Equations (3.114) can be solved analogously to (3.105,3.108) using orthogonal projections with appropriate norms as

$$ {}^{Q}\boldsymbol{p}_{nonfix} = \boldsymbol{Q}\nabla h^T \left[\langle \nabla h, \nabla h\rangle_{\boldsymbol{M}_Q^{-1}}\right]^{-1} \nabla h \boldsymbol{M}_Q^{-1}\boldsymbol{p}, \text{and} \tag{3.114a} $$

$$ {}^{P}\boldsymbol{p}_{nonfix} = \boldsymbol{P}^T\nabla h^T \left[\left\langle {}^{P}\nabla h^T, {}^{P}\nabla h^T\right\rangle_{\boldsymbol{M}_{red}^{-1}}\right]^{-1} \nabla h \boldsymbol{P}\boldsymbol{M}_{red}^{-1}\, {}^{P}\boldsymbol{p}. \tag{3.114b} $$

What remains is to distinguish the sliding component of the momentum nonfixed momentum from the normal component. This amounts to removing the normal momentum components from the nonfixed velocity, giving

$$ \boldsymbol{p}_{slide} = \boldsymbol{p}_{nonfix} - \boldsymbol{p}_{norm}. \tag{3.115} $$

In the presence of frictional forces, the post collision momenta will have the form

$$ {}^{Q}\boldsymbol{p}_{c^+}^{-} = {}^{Q}\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}^{+} + \boldsymbol{I}_{norm}({}^{Q}\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}^{+}) + \boldsymbol{I}_{slide,Q}({}^{Q}\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}^{+}) \tag{3.116a} $$

$$ {}^{P}\boldsymbol{p}_{c^+}^{-} = \bar{\boldsymbol{F}}_P\, {}^{P}\tilde{\boldsymbol{p}}_{c^+}^{+} + \boldsymbol{I}_{norm}(\bar{\boldsymbol{F}}_P\, {}^{P}\tilde{\boldsymbol{p}}_{c^+}^{+}) + \boldsymbol{I}_{slide,P}(\bar{\boldsymbol{F}}_P\, {}^{P}\tilde{\boldsymbol{p}}_{c^+}^{+}). \tag{3.116b} $$

A frictional impulse which captures slip-stick frictional behavior of Coulomb friction can be stated as follows. To begin, the maximum frictional impulse is given by $\boldsymbol{I}_{slide}^{max}(\boldsymbol{p}) = -\boldsymbol{p}_{slide}$, which corresponds to perfect stick. In the Coulomb model with friction coefficient, μ, the tangential forces are applied according to the ratio between normal and tangential forces, which can be approximated by assuming that the contact takes place over one time step, h, so that $h\boldsymbol{f}_{fric} = \boldsymbol{I}(\boldsymbol{p})$. In the extension of the key result of [15] to the present case of constrained rigid body dynamics, the sliding impulse can be defined as

$$ \boldsymbol{I}_{slide,Q}(\boldsymbol{p}) = \begin{cases} -\,{}^{Q}\boldsymbol{p}_{slide}, & \dfrac{\langle {}^{Q}\boldsymbol{p}_{slide}, {}^{Q}\boldsymbol{p}_{slide}\rangle_{M^{-1}}}{\langle {}^{Q}\boldsymbol{p}_{norm}, {}^{Q}\boldsymbol{p}_{norm}\rangle_{M^{-1}}} \le \mu \\ -\mu\dfrac{\langle {}^{Q}\boldsymbol{p}_{norm}, {}^{Q}\boldsymbol{p}_{norm}\rangle_{M^{-1}}}{\langle {}^{Q}\boldsymbol{p}_{slide}, {}^{Q}\boldsymbol{p}_{slide}\rangle_{M^{-1}}}\, {}^{Q}\boldsymbol{p}_{slide}, & \text{otherwise} \end{cases} \tag{3.117} $$

for the constrained formulation, or

$$ \boldsymbol{I}_{slide,P}(\boldsymbol{p}) = \begin{cases} -\,{}^{P}\boldsymbol{p}_{slide}, & \dfrac{\langle {}^{P}\boldsymbol{p}_{slide}, {}^{P}\boldsymbol{p}_{slide}\rangle_{M_{red}^{-1}}}{\langle {}^{P}\boldsymbol{p}_{norm}, {}^{P}\boldsymbol{p}_{norm}\rangle_{M_{red}^{-1}}} \le \mu \\ -\mu\dfrac{\langle {}^{P}\boldsymbol{p}_{norm}, {}^{P}\boldsymbol{p}_{norm}\rangle_{M_{red}^{-1}}}{\langle {}^{P}\boldsymbol{p}_{slide}, {}^{P}\boldsymbol{p}_{slide}\rangle_{M_{red}^{-1}}}\, {}^{P}\boldsymbol{p}_{slide}, & \text{otherwise} \end{cases} \tag{3.118} $$

for the reduced formulation. So, in the presence of Coulombic friction, the explicit momentum update reads

$$ {}^{Q}\boldsymbol{p}_{c^+}^{-} = {}^{Q}\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}^{+} + \boldsymbol{I}_{norm}({}^{Q}\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}^{+}) + \boldsymbol{I}_{slide,Q}({}^{Q}\bar{\boldsymbol{F}}_Q\tilde{\boldsymbol{p}}_{c^+}^{+}) \tag{3.119a} $$

$$ {}^{P}\boldsymbol{p}_{c^+}^{-} = \bar{\boldsymbol{F}}_P\,{}^{\tilde{P}}\tilde{\boldsymbol{p}}_{c^+}^{+} + \boldsymbol{I}_{norm}(\bar{\boldsymbol{F}}_P\,{}^{\tilde{P}}\tilde{\boldsymbol{p}}_{c^+}^{+}) + \boldsymbol{I}_{slide,P}(\bar{\boldsymbol{F}}_P\,{}^{\tilde{P}}\tilde{\boldsymbol{p}}_{c^+}^{+}). \tag{3.119b} $$

Multiple Collisions. Due to the assumption that collisions are simultaneous, accounting for multiple collisions amounts to using the gradient of $g(\boldsymbol{q}_c) = \sum_{i\in I_c} g_i(\boldsymbol{q})$ to resolve the normal momenta for each collision group, and solve one system for each group. Frictional forces, which depend on local relative motions, can be included by the sum of local impulses in the group.

3.5.5 DCR-DMC Algorithm Summary

In summary, the DCR-DCM algorithm is an explicit approximation to the exact algorithm developed in Section 3.4. The algorithm exactly conserves the total angular momentum components and the associated discrete Hamiltonian, H_d, in the sense that any non-physical changes in these quantities are due exactly to the closest point projection operation–or to intentional dissipation in the momentum updates. As such, collisions are accounted for in the algorithm in three simple steps:

1. Find an admissible configuration with CPP
2. Lift the momenta to the new subspace
3. Decompose and update momenta

with the momentum update depending on the type of collision being modeled.

Angular Momentum Check. While the orthogonal decomposition of subspaces necessarily leads to explicit momentum updates which conserve the total angular and linear momentum, as well as the discrete energy or Hamiltonian [‖], it is reassuring to directly check that the expected quantities are indeed preserved in the event that the approximation $\boldsymbol{q}_{c^+} \approx \tilde{\boldsymbol{q}}_{c^+}$ holds.

Because the equivalence between momentum decompositions and updates using the projected and reduced schemes has been established, we explore the relationship between conservation of discrete

[‖] Again, less any changes intentionally induced by the CPP operation or inelastic collisions.

quantities in the reduced scheme, and the expected conservation of the total linear and angular momentum of a system of rigid bodies. To begin, consider that in the momentum projection, only components of $\boldsymbol{p}_{c^+}^+ \in \eta(T^*_{\boldsymbol{q}_{c^+}}\mathcal{C}) \subseteq T^*_{\boldsymbol{q}_{c^+}}\mathcal{Q}$ are changed so that the update does not effect the the constraint Jacobian term, $\boldsymbol{G}(\boldsymbol{q}_{c^+})\boldsymbol{\lambda}_{c^+}$, which is computed in a post-processing step. So, it is only necessary to consider the effect of changes to ${}^{Q}\boldsymbol{p}_{c^+}$, or the equivalent changes to ${}^{P}\boldsymbol{p}_{c^+}$, to the total linear and angular momentum.

Recall the relationship between the velocities $\dot{\boldsymbol{\varphi}}^j$ and $\boldsymbol{\omega}^j$–with superscripts indicating the bodies to which these velocities are associated–and the projected redundant momenta given in (3.11) whereby

$$\boldsymbol{p}^{P,j} = \boldsymbol{MP}(\boldsymbol{q}^j)\boldsymbol{\nu}^j,$$

with $\boldsymbol{\nu}^j = \begin{bmatrix}\dot{\boldsymbol{\varphi}}^j & \boldsymbol{\omega}^j\end{bmatrix}^T$. The projected momenta, ${}^{P}\boldsymbol{p}^j \in \mathbb{R}^6$, can be written as

$${}^{P}\boldsymbol{p}^j = \boldsymbol{M}_{red}\boldsymbol{\nu}^j = \begin{bmatrix} m_{\varphi,j}\dot{\boldsymbol{\varphi}}^j \\ \boldsymbol{X}^j\boldsymbol{J}^j(\boldsymbol{X}^j)^T\boldsymbol{\omega}^j \end{bmatrix}.$$

As such, the total pre-collision linear and angular momentum, L^-_{lin} and L^-_{ang} due to the projected redundant momenta can be written as

$$\begin{aligned} \boldsymbol{L}^-_{lin} &= \sum_j m_{\varphi,j}\dot{\boldsymbol{\varphi}}^j \\ \boldsymbol{L}^-_{ang} &= \sum_j \left(\boldsymbol{\varphi}^j \times m_{\varphi,j}\dot{\boldsymbol{\varphi}}^j + \boldsymbol{X}^j\boldsymbol{J}^j{\boldsymbol{X}^j}^T\boldsymbol{\omega}^j\right). \end{aligned} \tag{3.120}$$

Furthermore, for the contact constraint gradient from the SSH LP, $\boldsymbol{P}^T\nabla g$ can be reduced to

$$\begin{aligned} \boldsymbol{P}^T\nabla_{\boldsymbol{q}^j} g &= \pm\boldsymbol{P}^T\boldsymbol{A}^j\boldsymbol{\alpha} \\ &= \begin{bmatrix} \pm\boldsymbol{\alpha} \\ \pm\boldsymbol{r}^j \times \boldsymbol{\alpha} \end{bmatrix}, \end{aligned}$$

and $\boldsymbol{P}^T(\nabla_{\boldsymbol{q}^j}\boldsymbol{h})^T$ also has the form

$$\begin{aligned}\boldsymbol{P}^T(\nabla_{\boldsymbol{q}^j}\boldsymbol{h})^T &= \pm\boldsymbol{P}^T\boldsymbol{A}^j\boldsymbol{\eta} \\ &= \begin{bmatrix} \pm\boldsymbol{\eta} \\ \pm\boldsymbol{r}^j \times \boldsymbol{\eta} \end{bmatrix},\end{aligned}$$

in which the sign simply depends on how the bodies involved in collisions were indexed, $\boldsymbol{r}^j = \sum_{i\in I_j} \pi_i \boldsymbol{v}_{i,I}^j \boldsymbol{d}_I^j$, represents the vector from the body's center of mass to the center of the contact surface, $\boldsymbol{\alpha}$ is normal to the contact surface, and

$$\boldsymbol{\eta} = \left[\left\langle {}^P\nabla h^T, {}^P\nabla h^T\right\rangle_{\boldsymbol{M}_{red}^{-1}}\right]^{-1} \nabla h \boldsymbol{P}\boldsymbol{M}_{red}^{-1}\, {}^P\boldsymbol{p}.$$

So that the total post-collision momentum can then be expressed in the form

$$\begin{aligned}\boldsymbol{L}_{lin}^+ &= \sum_j \left[m_{\varphi,j}\dot{\boldsymbol{\varphi}} \mp (\gamma_1\boldsymbol{\alpha} + \gamma_2\boldsymbol{\eta})\right] \\ \boldsymbol{L}_{ang}^+ &= \sum_j \left[\boldsymbol{\varphi}^j \times (m_{\varphi,j}\dot{\boldsymbol{\varphi}}^j \mp [\gamma_1\boldsymbol{\alpha} + \gamma_2\boldsymbol{\eta}]) + \boldsymbol{X}^j\boldsymbol{J}^j{\boldsymbol{X}^j}^T\boldsymbol{\omega}^j \mp (\boldsymbol{r}^j \times [\gamma_1\boldsymbol{\alpha} + \gamma_2\boldsymbol{\eta}])\right],\end{aligned}$$

with $\gamma_1, \gamma_2 \in \mathbb{R}$ depending on the type of collision being modeled. Thus the changes in total momentum are given by

$$\boldsymbol{L}_{lin}^+ - \boldsymbol{L}_{lin}^- = \sum_j (\pm\gamma_1\boldsymbol{\alpha} \pm \gamma_2\boldsymbol{\eta}) = \boldsymbol{0}$$

Due to the pairwise nature of collisions. The change in angular momentum is given by

$$\begin{aligned}\boldsymbol{L}_{ang}^+ - \boldsymbol{L}_{ang}^- &= \sum_j \left[\pm\boldsymbol{\varphi}^j \times (\gamma_1\boldsymbol{\alpha} + \boldsymbol{\gamma}_2\eta) \pm (\boldsymbol{r}^j \times [\gamma_1\boldsymbol{\alpha} + \gamma_2\boldsymbol{\eta}])\right] \\ &= \sum_j \left[\pm(\boldsymbol{\varphi}^j + \boldsymbol{r}^j) \times (\gamma_1\boldsymbol{\alpha} + \gamma_2\boldsymbol{\eta})\right] \\ &= \sum_j \left[\boldsymbol{0} \times (\gamma_1\boldsymbol{\alpha} + \gamma_2\boldsymbol{\eta})\right] \\ &= \boldsymbol{0},\end{aligned}$$

because $\boldsymbol{h} = \boldsymbol{\varphi}^1 - \boldsymbol{\varphi}^2 + \boldsymbol{r}^1 - \boldsymbol{r}^2 = \boldsymbol{0}$ for each collision due to the KKT conditions at the optimum of the SSH LP.

3.5.6 Examples

In the following examples, the DCR-DMC integrator developed in the preceding section is used in the $(\boldsymbol{q}, \boldsymbol{p})$ formulation and the reduced discrete Legendre transform, along with the appropriate local parametrization of the update for each example. The state of the system is advance in the normal way using the constrained discrete variational integrator until overlap is detected. Once detected, the overlapping configuration is thrown out and the an admissible configuration is found using closest-point projection, and norms in the updated subspace are used to decomposed and explicitly update the state of the system to the post-collision state. Collisions between polyhedral bodies are detected using the SSH LP, which is also used to determine $\nabla g(\boldsymbol{q}_{c^+})$ and $\nabla h(\boldsymbol{q}_{c^+})$ for use in the momentum decompositions. In all examples, the energy shown is associated with the discrete Hamiltonian,

$$H_d(\boldsymbol{q}_k, \boldsymbol{p}_k) = \frac{1}{2}\boldsymbol{p}_k^T \boldsymbol{M}^{-1} \boldsymbol{p}_k + \boldsymbol{V}(\boldsymbol{q}_k),$$

with the momenta $\boldsymbol{p}_k$ given by the constrained discrete Legendre transform. Changes in energy induced by the CPP operation are calculated in the obvious way as

$$\Delta V_{cpp} = V(\boldsymbol{q}) - V(\tilde{\boldsymbol{q}})$$

and changes to the total angular momentum due the CPP operation can be calculated from the projected redundant momenta as

$$\Delta \boldsymbol{L}_{cpp} = \boldsymbol{\varphi} \times^{Q} \boldsymbol{p}_{\varphi} + \boldsymbol{d}_I \times^{Q} \boldsymbol{p}_I - \left(\tilde{\boldsymbol{\varphi}} \times^{\tilde{Q}} \tilde{\boldsymbol{p}}_{\varphi} + \tilde{\boldsymbol{d}}_I \times^{\tilde{Q}} \tilde{\boldsymbol{p}}_I \right),$$

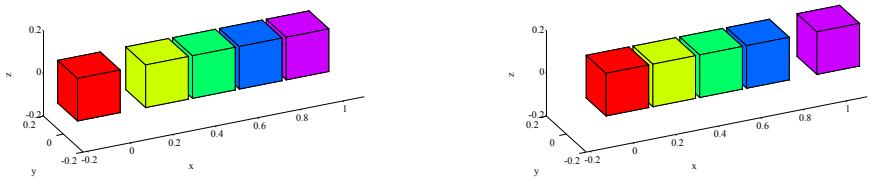
in which all $\tilde{(\cdot)}$ values are associated with the inadmissible state $(\tilde{\boldsymbol{q}}, \tilde{\boldsymbol{p}})$, and all other values are given by the mappings $\boldsymbol{q} = \boldsymbol{F}_{cpp}(\tilde{\boldsymbol{q}})$ and $\boldsymbol{p} = \bar{\boldsymbol{F}}_Q \cdot \tilde{\boldsymbol{p}}$.

In the contact intervals, Matlab's built in nonlinear constrained optimization function, fmincon, is used, with the sequential quadratic programming algorithm to solve a series of local CPP problems

defined by (3.82). Even though the contact potential is used to constrain the CPP problem, the SSH LP is used directly without any modifications in subsequent steps to accurately decompose and update the momentum. The examples parallel the examples shown for the constrained mechanical integrator in Sections 3.3 and 3.4, with additional examples demonstrating systems with inelastic collisions.

'Newton's Cradle'. The dynamics of the well-known Newton's Cradle toy (see Figure 3.20**) modified so that the balls are replaced by cubes, and the motion of the bodies is unconstrained (i.e., the strings are not modeled so there is only one series of collisions) are investigated for perfectly elastic collisions.

Two cases are considered. In case 1, four cubes are aligned at rest of side length $l = 0.2\sqrt{2}$ m and uniform density $\rho = 270$ $^{\text{kg}}/_{\text{m}^3}$ and separated by $l/10$. One cube of the same dimension starts with its face a distance of $l/2$ from the face of the end cube and moves toward the bunch with initial velocity $\dot{\boldsymbol{\varphi}}^1 = [0.105\ 0\ 0]^T$ $^{\text{m}}/_{\text{s}}$. As can be seen in the images in Figure 3.66, the impulse of the initial impact leads to the cube on the opposite end of the bunch from the original moving cube departing the cluster. The evolution of the energy and total angular momentum for this case are shown in Figure 3.67, with the actual angular momentum and the angular momentum with the changes due to the CPP operation removed also shown. The fixed time step is $h = 0.015$.



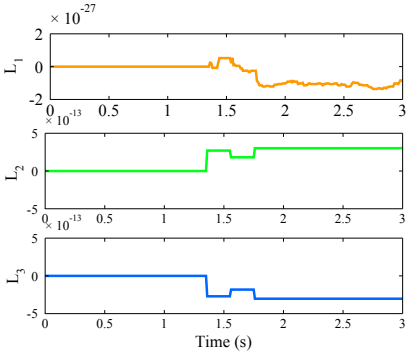
(a) t = 0s (b) t = 3s

Figure 3.66: Initial and final configurations for first case of 'Newton's Cradle' examples.

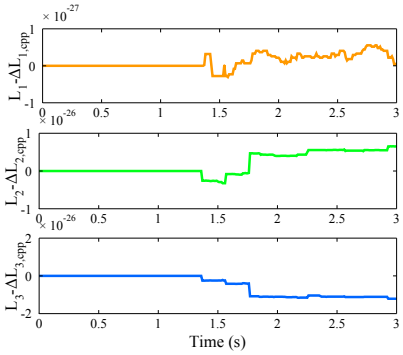
In case 2, three cubes are aligned at rest of side length $l = 0.2\sqrt{2}$ m and uniform density $\rho = 270$ $^{\text{kg}}/_{\text{m}^3}$ and separated by $l/100$. Two cubes of the same dimension start with faces a distance of $l/2$ from the face of the end cubes and move toward the bunch with initial velocities $\dot{\boldsymbol{\varphi}}^1 = [0.105\ 0\ 0]^T$ $^{\text{m}}/_{\text{s}}$ and $\dot{\boldsymbol{\varphi}}^5 = [-0.105\ 0\ 0]^T$ $^{\text{m}}/_{\text{s}}$, respectively. As can be seen in the images in Figure 3.68, the impulse of the initial impact leads to both of the end cubes moving away from the cluster. The evolution of the energy and angular momentum for this case are shown in Figure 3.69.

In both of these examples, no relative rotation between the bodies is induced by the use of a the SSH LP to define $\nabla g(\boldsymbol{q}_{c^+})$, which is the expected result for perfectly flat face-face contact. Before

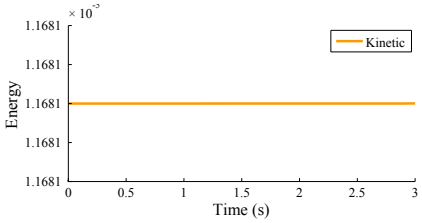
**Which is now extremely well known to close readers of this thesis.



(a) Angular Momentum.

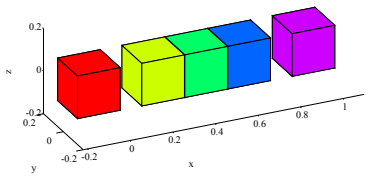


(b) Angular Momentum Less $\Delta \boldsymbol{L}_{cpp}$.

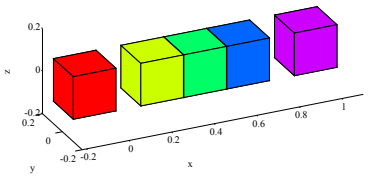


(c) Energy.

Figure 3.67: Angular momentum and energy evolution for the first case of 'Newton's Cradle' examples for $h = 0.01$s. In the plot to the right of the angular momentum, changes in $\boldsymbol{L}$ due to the CPP operation are removed.



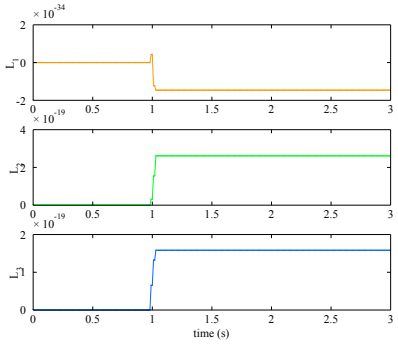
(a) t = 0s



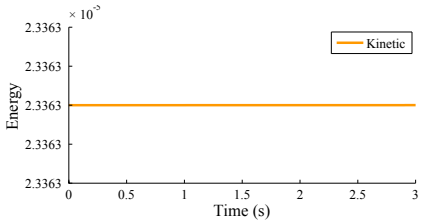
(b) t = 2.2s

Figure 3.68: Initial and final configurations for second case of 'Newton's Cradle' examples.

proceeding to more involved examples, two cases of perfectly plastic collisions between two cubes are treated. In the first perfectly plastic contact example, one cube moves towards a stationary cube with a speed of 0.5m⁄s. In the second perfectly plastic contact example, both cubes move towards each other with a speed of 0.5m⁄s, so that the relative speed is 1.0m⁄s. Several time lapse images of the motion, as well as the evolution of the energy are shown in Figures 3.70 and 3.71, as well as Figure 3.72, respectively, and correspond to the expected results from a simple algebraic calculation.

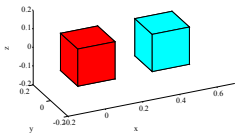


(a) Angular Momentum.

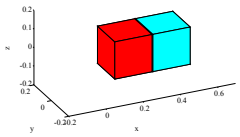


(b) Energy.

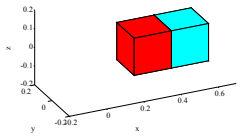
Figure 3.69: Angular momentum and energy evolution for and second case of 'Newton's Cradle' examples for $h = 0.01$s.



(a) t = 0s

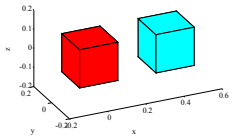


(b) t = 0.4s

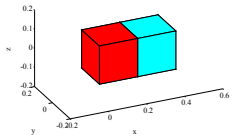


(c) t = 0.8s

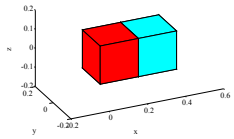
Figure 3.70: Time lapse images of a perfectly plastic collision between on moving (red) and one stationary cube (blue). After the collision, both cubes move together with half of the initial velocity.



(a) t = 0s

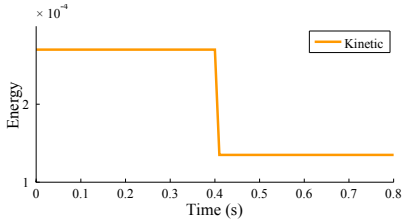


(b) t = 0.2s

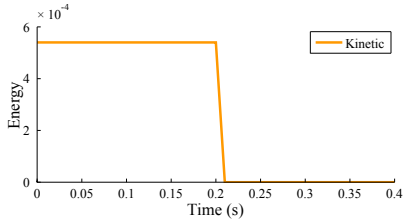


(c) t = 0.4s

Figure 3.71: Time lapse images of a perfectly plastic collision between two moving cubes. After the collision, the cubes remain stationary.



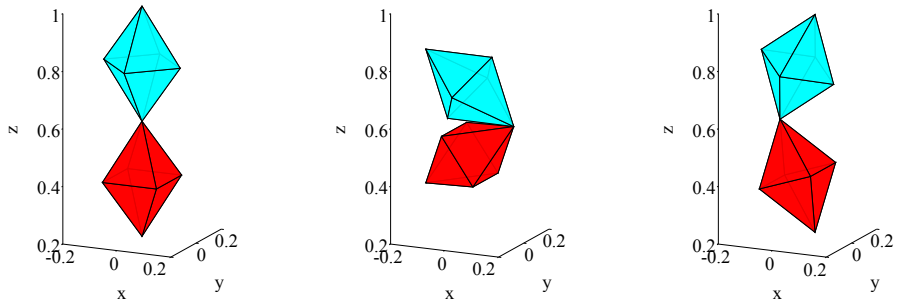
(a) Energy, $v_{rel} = 0.5$m⁄s (one moving cubes).



(b) Energy, $v_{rel} = 1.0$m⁄s (two moving cube)

Figure 3.72: Energy evolution for perfectly plastic 'Newton's Cradle' examples for $h = 0.01$s.

Metronome. As in the analogous potential-based and exact examples, a simple kinematic chain consisting of two octohedral prisms with side length $0.2\sqrt{3}$ m connected by a spherical joint at the top of the first body and the bottom of the second body is introduced. In the initial configuration, $\boldsymbol{d}_3^1 = \boldsymbol{d}_3^2$, and the remaining directors of the second body are rotated by $\frac{\pi}{4}$rad about $\boldsymbol{d}_3$ relative to the first. The initial velocities used to start the simulation are $\boldsymbol{\omega}_1 = [0\ 1\ 0]^T$ rad⁄s and $\boldsymbol{\omega}_2 = [0\ -1\ 0]^T$ rad⁄s, so that the bodies rotate about the joint and swing towards each other, collide when the edge of the second body strikes the face of the first body, and so on. Several time lapse images of this motion are shown in Figure 3.73. The evolution of the energies (in terms of ${}^{Q}H_d$ for this example) and angular momentum for the metronome example using the DCR-DMC algorithm are shown in Figure 3.74. In the example shown, the fixed time step is $h = 0.015$s, and (clearly) only pairwise collisions take place.

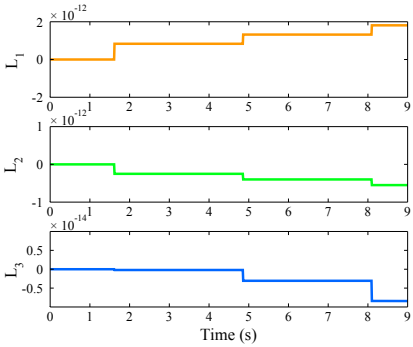


(a) $t = 0.0$s (b) $t = 1.8$s (c) $t = 3.6$s

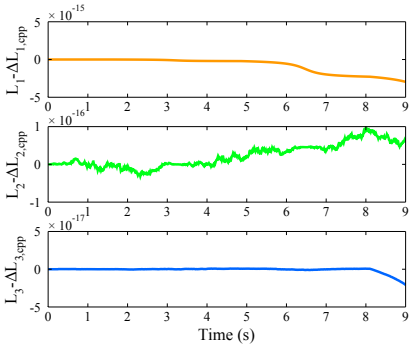
(d) $t = 5.4$s (e) $t = 7.2$s (f) $t = 9.0$s

Figure 3.73: Time lapse images for a prismatic metronome connected by a spherical joint using the DCR-DMC algorithm.

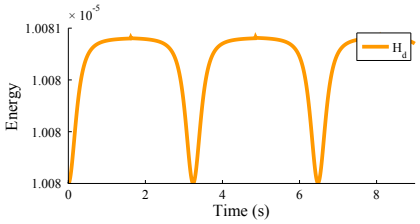
As in the Newton's cradle examples, no additional relative rotation is induced by the use of the SSH LP's gradient to decompose and update the momenta, which is the expected result for this particular edge-face contact configuration. As shown in Figure 3.74b, removing the changes in angular momentum due to the CPP operation leads to the conservation results as the exact algorithm.



(a) Angular Momentum.



(b) Angular Momentum less $\Delta \boldsymbol{L}_{cpp}$.



(c) Energy.

Figure 3.74: Angular momentum and energy evolution for a prismatic metronome connected by a spherical joint solved for $h = 0.015$s. The CPP portion of the DCR-DMC algorithm causes small changes in otherwise conserved components of the angular momentum.

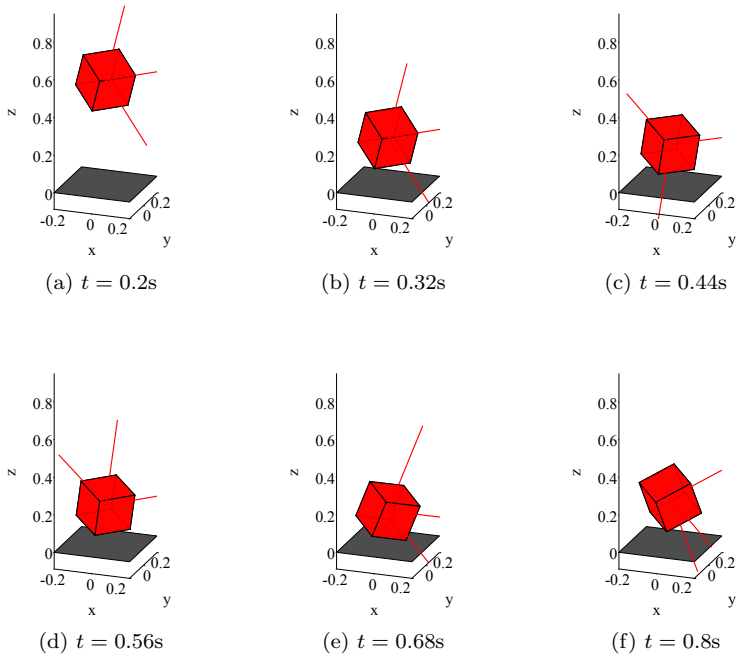
Falling Polyhedra. The simple case of a cube colliding with a floor under the influence of a gravitational potential

$$V_g(\boldsymbol{q}) = -g\boldsymbol{e}_g^T \boldsymbol{M}\boldsymbol{q}$$

with $g = 9.81\text{m/s}^2$ and $\boldsymbol{e}_g^i = [\boldsymbol{e}_z \; \mathbf{0} \; \mathbf{0} \; \mathbf{0}]^T \in \mathbb{R}^{12}$ is investigated first in order to compare the results when the potential is (as in (3.83)) or is not (as in (3.82)) included in the CPP update, which are essentially indistinguishable. This example differs from previous examples due to the presence of a potential; now the CPP operation will also effect the overall energy of the system and not just the momentum. The cube with properties described below is dropped with zero initial velocity from height of $\boldsymbol{\phi} = 0.8\text{m}$. Time lapse images of the motion for $t \in [0.2, 0.8]\, s$ are shown in Figure 3.75. The evolution of the energy and angular momentum solved using (3.82) is shown in Figure 3.76 and the same plots using (3.82) are shown in Figure 3.77; note that if changes due to the projection operation are removed, L_3 is exactly preserved. The results using the CPP problem with the potential energy, (3.83), show marginally smaller changes to the conserved quantities, but there is not a significant difference. The small incremental increases in the total energy are also due to the CPP operation, which has the effect of instantaneously increasing the gravitational potential without effecting the kinetic energy, thus causing a small increase in the overall energy. In this example, $h = 0.001\text{s}$ although a larger time step could be taken depending on the needed accuracy. The collisions are perfectly elastic ($e = 1, \mu = 0$).

In the analog to the previous examples of falling polyhedra, seven polyhedra start in a flower-like cluster with centers of mass $\sim 0.8\text{m}$ above a 'floor' occupying the $x - y$ plane, with a small $\sim 0.01\text{m/s}$ random initial translational velocity, and no initial angular velocity. For the purposes of this example, the floor is also modeled as a polyhedron, with vertices at $(x, y) = (\pm 10, \pm 10)\text{cm}$, and normal $\boldsymbol{e}_z$. In the evaluation of the SSH LP to determine overlap and ∇g for body-floor interactions, $\hat{\boldsymbol{\beta}} = \boldsymbol{e}_z$ is always used. Three different shapes are used; a cube with side length 0.2m, a right square pyramid with base dimension 0.2m and height 0.4m, and a regular octohedral prism with side length $0.2\sqrt{3}\text{m}$, all with a uniform density of $\rho = 270\text{kg/m}^3$. A series of time lapse images of the system is shown in Figure 3.78, and the evolution of the total energy and angular momentum is shown in Figure 3.79. Several examples of multiple collisions are present, which are dealt with seamlessly by the algorithm.

Before moving on, we return to the simple system of the cube hitting the floor, but now model the collisions as inelastic $e = 0.6$ and frictional $\mu = 0.8$. Because energy is lost from the system at

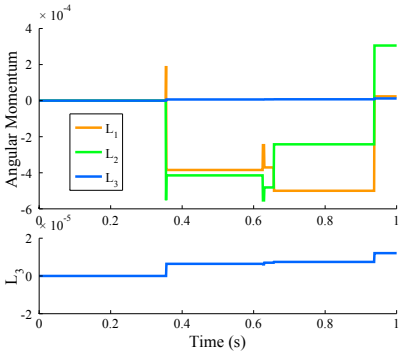


(a) $t = 0.2$s (b) $t = 0.32$s (c) $t = 0.44$s

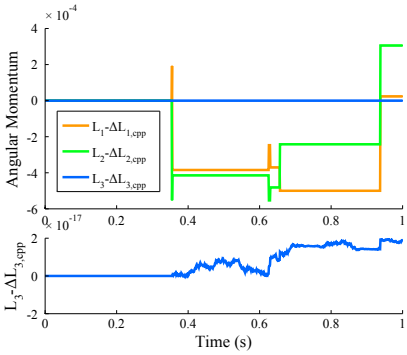
(d) $t = 0.56$s (e) $t = 0.68$s (f) $t = 0.8$s

Figure 3.75: Time lapse images for a falling cube under the influence of gravity.

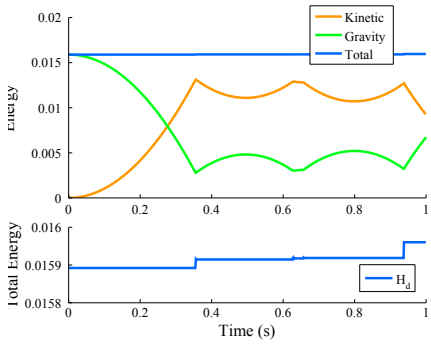
every collision, a 'static' solution with the cube 'resting' on the floor is eventually reached, as shown in Figure 3.80. Even though the DCR-DMC algorithm is intended as a collision integrator, the algorithm does a reasonable job dealing with an essentially static problem, with the kinetic energy tending to zero and the gravitational energy attaining a minimum. Note that the presence of friction negates the conservation of L_3 in the initial series of collisions, but the angular momentum oscillates about $\boldsymbol{L} = \mathbf{0}$, which is the expected result. The evolution of the energy and momentum less changes due to the CPP operation are shown in Figure 3.81.



(a) Angular Momentum.

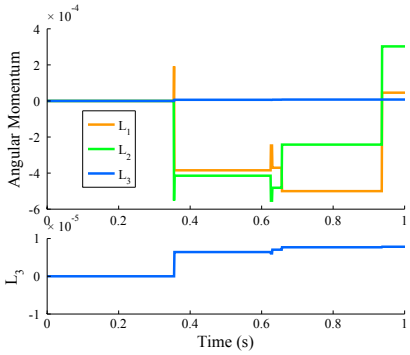


(b) Angular Momentum Less $\Delta \boldsymbol{L}_{cpp}$.

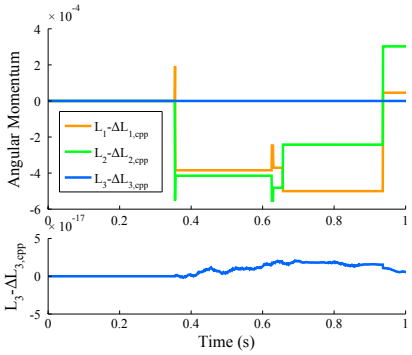


(c) Energy.

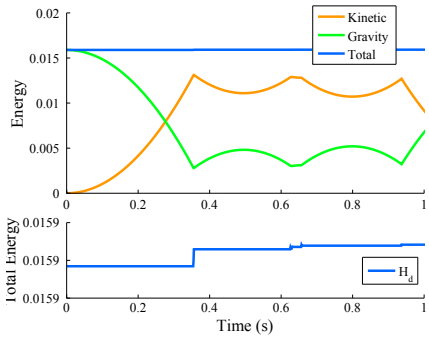
Figure 3.76: Angular momentum and energy evolution for a bouncing cube with $h = 0.001$s and (3.82) which does not include the potential energy term used to resolve $\boldsymbol{q}_{c^+} = F_{cpp}(\tilde{\boldsymbol{q}}_{c^+})$. In the plot to the right of the angular momentum, changes in $\boldsymbol{L}$ due to the CPP operation are removed. The incremental increase in energy is also due to the CPP operation, which has the effect of increasing the potential energy in the system.



(a) Angular Momentum.

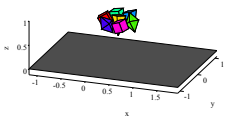


(b) Angular Momentum Less $\Delta \boldsymbol{L}_{cpp}$.

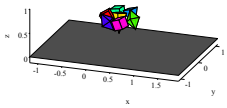


(c) Energy.

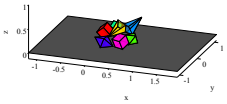
Figure 3.77: Angular momentum and energy evolution for a bouncing cube with $h = 0.001$s and (3.83) which includes the potential energy term used to resolve $\boldsymbol{q}_{c^+} = F_{cpp}(\tilde{\boldsymbol{q}}_{c^+})$. In the plot to the right of the angular momentum, changes in $\boldsymbol{L}$ due to the CPP operation are removed. The incremental increase in energy is also due to the CPP operation, which has the effect of increasing the potential energy in the system.



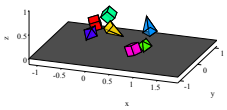
(a) $t = 0$s



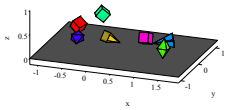
(b) $t = 0.2$s



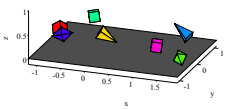
(c) $t = 0.4$s



(d) $t = 0.6$s

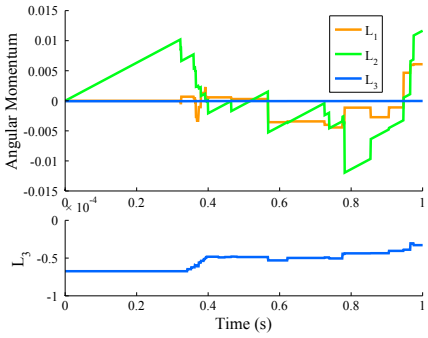


(e) $t = 0.8$s

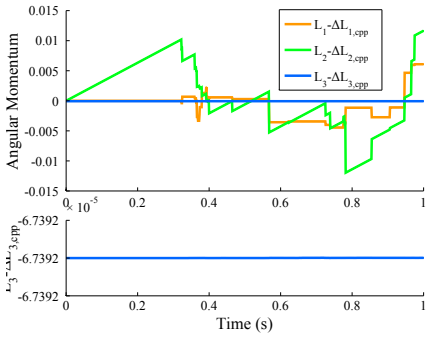


(f) $t = 1.0$s

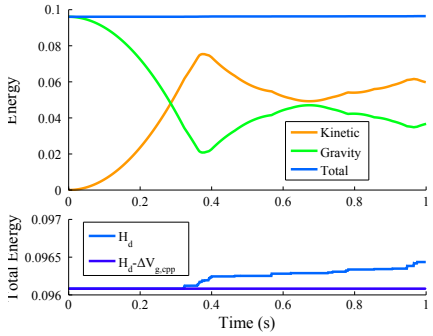
Figure 3.78: Time lapse images for falling polyhedra using the DCR-DMC algorithm.



(a) Angular Momentum.

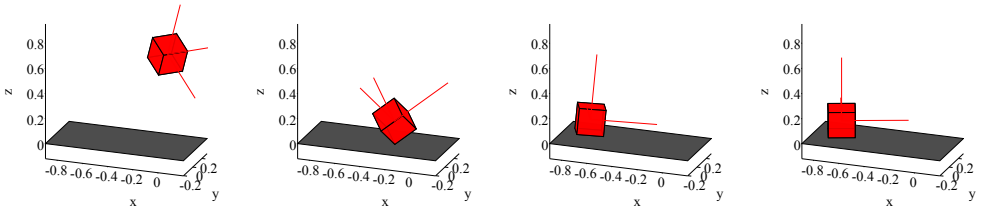


(b) Angular Momentum less $\Delta \boldsymbol{L}_{cpp}$.



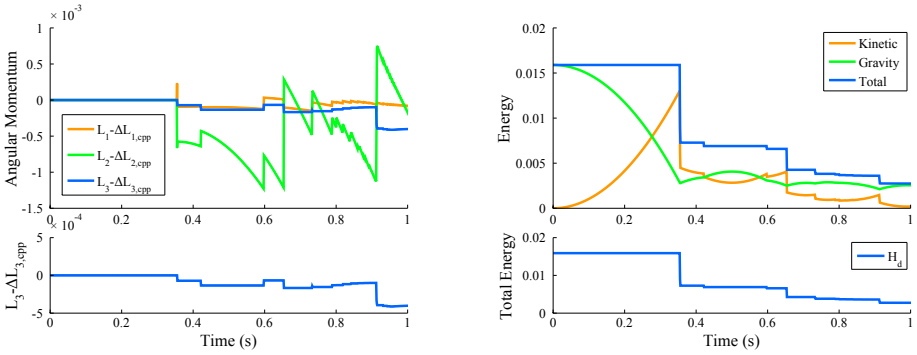
(c) Energy.

Figure 3.79: Angular momentum and energy evolution for falling polyhedra with $h = 0.001$s. Small changes in conserved quantities are due to the CPP operation.



(a) $t = 0$s (b) $t = 0.5$s (c) $t = 1.0$s (d) $t = 1.5$s

Figure 3.80: Time lapse images for a cube falling onto an inelastic frictional floor using the DCR-DMC algorithm.



(a) Angular Momentum less $\Delta \boldsymbol{L}_{cpp}$.

(b) Energy.

Figure 3.81: Angular momentum and energy evolution for a cube falling on an inelastic, frictional floor with $e = 0.6$, $\mu = 0.8$ and $h = 0.001$s. Small changes in conserved quantities are due to the CPP operation or to the frictional interactions between the cube and the floor.

Hexagonal Prisms. This example illustrates an instance in which the DCR-DMC algorithm overcomes the limitations of the discrete variational exact collision integrator in Section 3.4. In addition, by the introduction of dissipation into the system via friction or restitution, the formerly 'potentially' clumping rigid right hexagonal prisms successfully assemble into a coherent structure. The prisms have height 0.4m and a side length of 0.2m. As in previous examples, a series of alternating positive and negative Coulombic charges are centered behind each side face (see Figure 3.82), with the potential given in (3.51). In the present example, the constant $K = 5e - 4$.

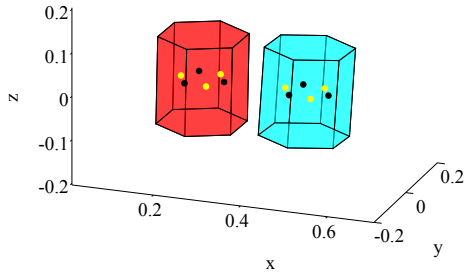
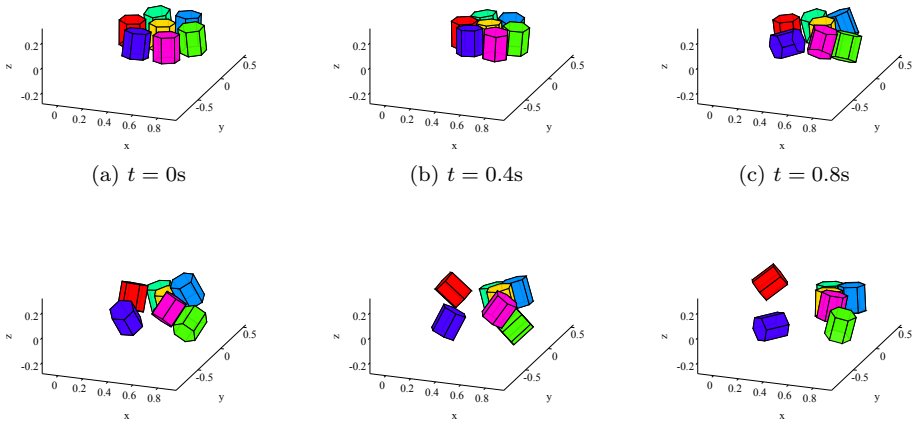


Figure 3.82: Hexagonal prisms shown with locations of alternating positive and negative (yellow and black) Coulombic charges.

This type of system has several complicating factors for the exact integrator which are overcome by the DCR-DMC integrator. For the exact integrator, aside from the obvious issues of multiple contacts being compounded by the attractive potentials, the potential depends on the *relative* configuration of each pair of bodies, so that the discrete collision equations must be solved for the full system, so that the impact equations could not be reduced to local problems. In the DCR-DMC algorithm, these issues do not pose a problem because the basis of the algorithm is the assumption of simultaneous collisions at the end of a times step. That being said, as in the example with a gravitational potential, the CPP operation causes an in increase in the overall energy of the system if no dissipation is present, as can be seen in Figure 3.84, which also shows the evolution of the angular momentum with changes due the CPP operation removed. Time lapse images of the motion for a time step of $h = 1e - 3$s are shown in Figure 3.83.

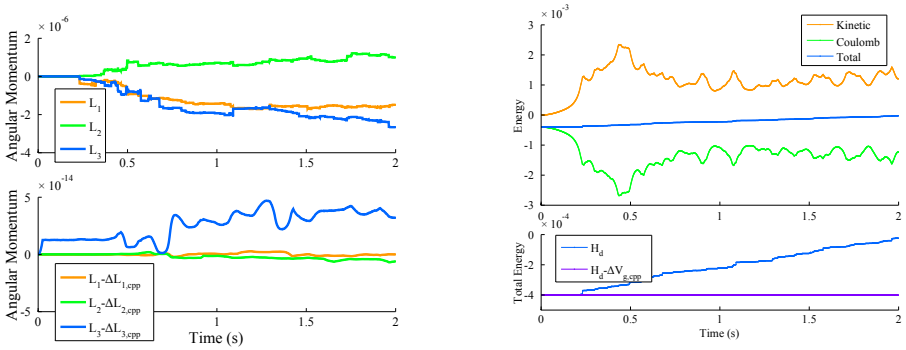
By making all collisions frictional with $\mu = 0.8$ and inelastic with $e = 0.8$, the DCR-DCM algorithm successfully captures the assembly of the hexagons into a coherent flower like structure with only small oscillations about the minimum energy, as shown in Figures 3.85 and 3.86.



(a) $t = 0$s (b) $t = 0.4$s (c) $t = 0.8$s

(d) $t = 1.2$s (e) $t = 1.6$s (f) $t = 2.0$s

Figure 3.83: Time lapse images for rigid hexagonal prisms with the motion driven by internal Coulombic charges, as shown in Figure 3.82. No dissipation is present in this system.

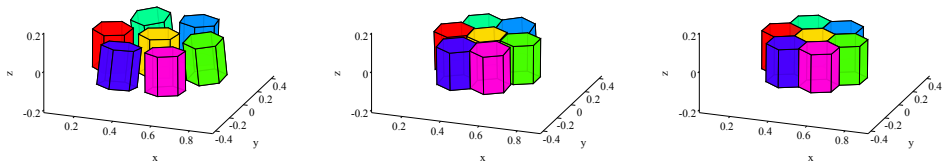


(a) Angular Momentum. (b) Energy.

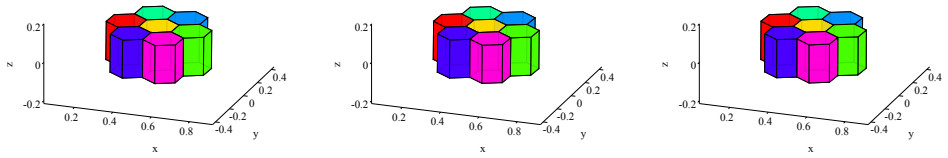
Figure 3.84: Angular momentum and energy evolution for rigid hexagonal prisms with $h = 0.001$s and no dissipation. The incremental increase in energy is due to the CPP operation, which increases the Coulombic potential energy.

3.5.7 Discussion

This section has developed the DCR-DMC algorithm as an approximation to the exact algorithm in Section 3.4, with collision detection and explicit momentum updates enabled by the SSH LP. For perfectly elastic collisions, the DCR-DCM algorithm *approximately* conservers the total energy and the appropriate components of the total momenta in the system, with small changes due exclusively to the CPP operation, and is thus deemed a conserving approximation. Furthermore, taking a decomposition approach to resolving collision allows the DCR-DCM algorithm to overcome issues that render the exact algorithm prohibitively expensive due to multiple collisions in close succession or relative configuration dependent potentials. If dissipation is present, the algorithm can successfully



(a) $t = 0$s (b) $t = 0.4$s (c) $t = 0.8$s



(d) $t = 1.2$s (e) $t = 1.6$s (f) $t = 2.0$s

Figure 3.85: Time lapse images for rigid hexagonal prisms with the motion driven by internal Coulombic charges, as shown in Figure 3.82. The collisions are modeled as inelastic and frictional with $\mu = 0.8$ and $e = 0.8$. The steady state solution at a minimal potential energy is successfully captured.

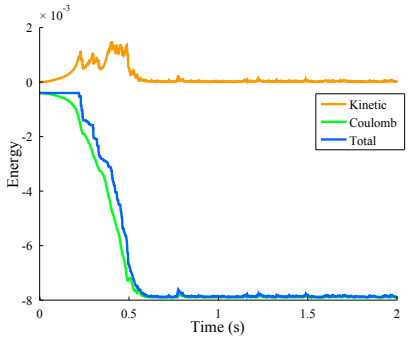


Figure 3.86: Energy evolution for rigid hexagononal prisms with $h = 0.001$s with $\mu = 0.8$ and $e = 0.8$. The solution approaches a steady state with only small oscillations about the minimum.

approximate the steady state (minimal energy) solution, with only small oscillations about the minimum if the minimum occurs in a contact configuration. The main drawbacks of the algorithm are the overall increase or decrease in energy if a potential is present in the system, because the CPP operation has the effect of changing the potential energy in the system. This effect, along with the expected changes in angular momentum, can be mitigated by decreasing h, depending on the desired accuracy of the simulation.

3.6 Discussion

Three related collision integrators for constrained mechanical systems have been introduced in this section; a parameter-dependent potential-based approach, an exact integrator, and the decomposition-based DCR-DCM algorithm, which is a conserving approximation. Each of these approaches has advantages and drawbacks, as summarized in Table 3.1.

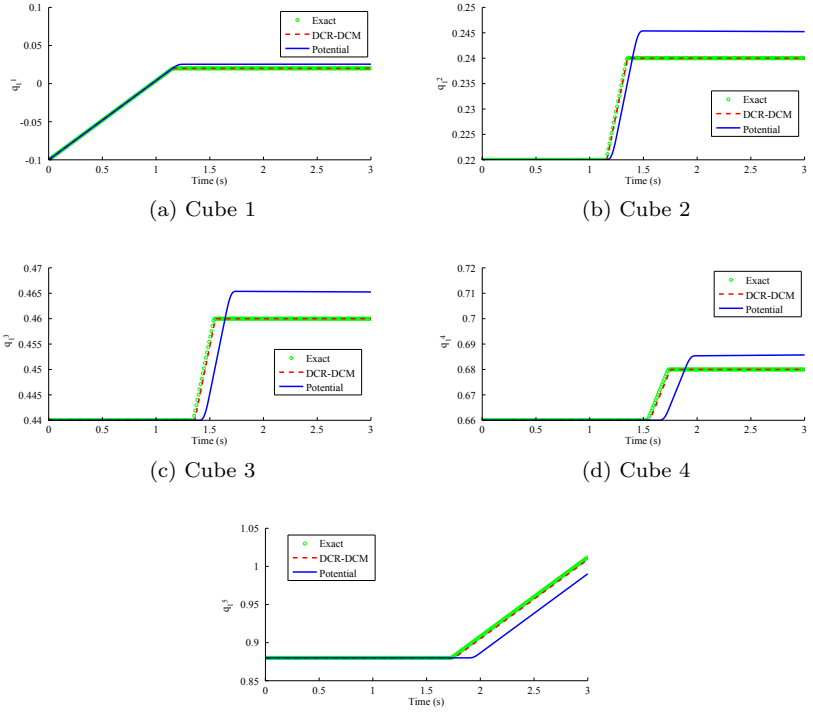
Algorithm	**Advantages**	**Disadvantages**	**Comments**
Potential	Straightforward implementation, especially for lumped mass finite element models	Parameter dependence limits fidelity and time step	Fidelity can be improved by increasing parameter and decreasing time step
Exact	Exactness	Algorithm prohibitively expensive for multiple collisions and relative configuration dependent potentials	There is no good way to ameliorate the 'cons', Restitution could be added to model inelastic collisions, but this would add to the existing problems with multiple collisions
DCR-DCM	Known conservation properties, No issues for multiple collisions, Readily models dissipative systems	Inexact conservation due to CPP, particularly for energy when potential present	Conservation properties can be improved by decrease in time step, straightforward and relatively inexpensive simulation of self-assembling systems

Table 3.1: Comparison between the algorithms introduced in Sections 3.3, 3.4, and 3.5.

The same four examples were run (or attempted) for each integrator. The final two examples (falling polyhedra and clumping hexagonal prisms) do not lend themselves to direct comparison due to their chaotic behavior, however the regular motions in the first two examples (Newton's cradle and prismatic metronome) do.

Newton's Cradle. As shown in Figures 3.87 and 3.88, which track the $x-$position of each cube in the series, the DCR-DCM solutions to both cases of the Newton's cradle examples closely follow the solution from the exact discrete variational collision integrator. The prolonged overlap allowed by the potential-based approach leads to a quantitative difference in the motion of the cubes. However, the *qualitative* difference in the observed motions is small, and may be acceptable given the ease of implementation of the potential-based approach.

Prismatic Metronome. As shown in Figure 3.89, which track the $x-$position of the joint location, the DCR-DCM solution closely follow the exact solution, however the loss of energy in the



(a) Cube 1

(b) Cube 2

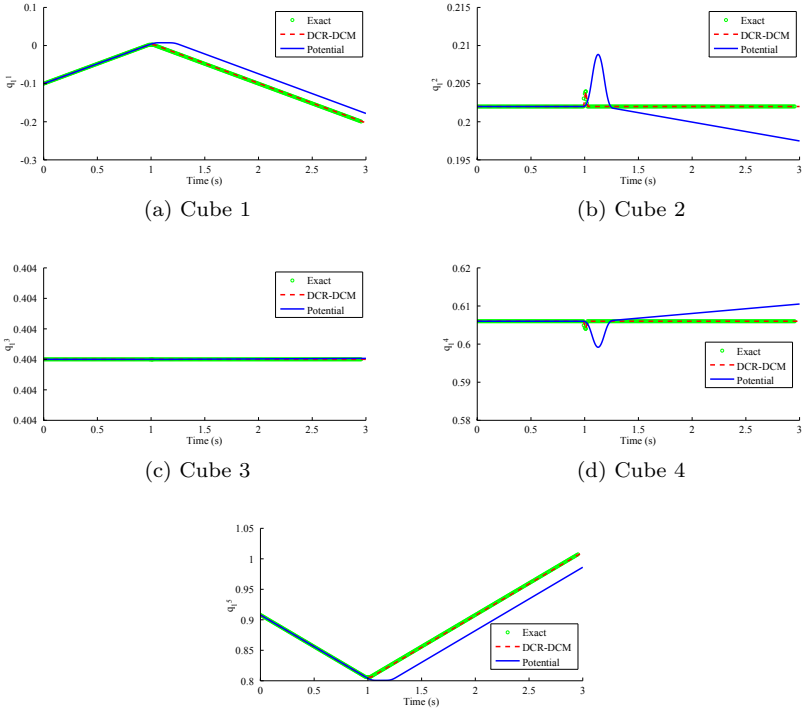
(c) Cube 3

(d) Cube 4

(e) Cube 5

Figure 3.87: $x-$positions of the Newton's cradle cubes for the first case in which there are no multiple collisions. Cubes are numbered in terms of their initial $x-$position in the global frame.

potential-based solution as discussed in Section 3.3.5 (plotted for $h = 1e-4$s) is clearly present. Figure 3.90 compares the the value of ${}^{Q}H_d$ for the DCR-DCM and exact schemes, which are essentially indistinguishable, and shows that the dissipation dominates the behavior of the potential scheme. This dissipation is even more pronounced at larger time steps, and can be improved by further decreasing the time step.



(a) Cube 1

(b) Cube 2

(c) Cube 3

(d) Cube 4

(e) Cube 5

Figure 3.88: $x-$positions of the Newton's cradle cubes for the second case in which there are multiple collisions. Cubes are numbered in terms of their initial $x-$position in the global frame.

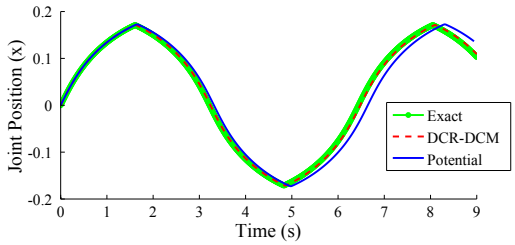
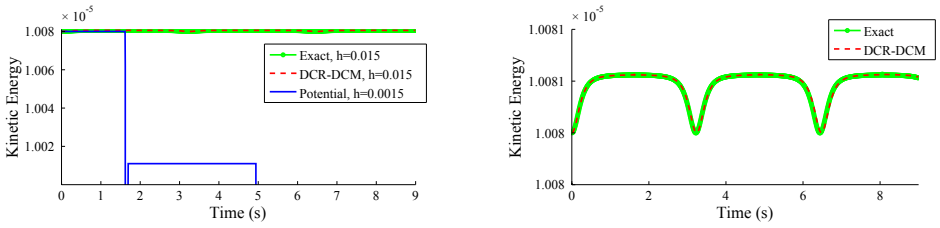


Figure 3.89: Joint $x-$location for the Exact, DCR-DCM, and Potential–based schemes.



(a) H_d, All Schemes

(b) H_d, Exact and DCR-DCM

Figure 3.90: Energy calculated from the restricted discrete Hamiltonian, H_d for all collision integrators.

3.7 Applications: Robust Docking System Design

This section showcases the use of the collision integrators to the design of a kinematic docking system for CubeSats docking, and, more specifically to the preliminary design of the autonomous docking system for the AAReST mission (see Figure 1.1). The AAReST mission itself is a collaborative effort between a multidisciplinary team at Caltech, the University of Surrey, and the Jet Propulsion Laboratory (JPL), with the goal of demonstrating various components for a self-assembling space telescope using a low cost CubeSat platform, including deformable mirrors (c. f. Patterson et al. [70]) and autonomous on-orbit reconfiguration. While an exact launch date has not been determined, the tentative date is late in 2014.

For the purposes of this section, it will be convenient to call the central cluster of CubeSats the 'base' satellite, and the CubeSat involved in the reconfiguration the 'mirrorcraft'. The on-orbit reconfiguration maneuver can be considered in three parts. In the first part, the docking system must release the mirror craft from the base satellite; in the second, a chemical propulsion system* is used to move the mirrorcraft into alignment with a different set of docking ports; in the third and final stage, which is the primary concern of the design problems in this section, a magnetic actuation system† is used to perform the final alignment and capture of the mirrorcraft into its new position.

This section is concerned with the design of various components for the docking and latching system, including the mechanical design of the docking ports. These problems are treated with a combination of the computational tools developed in this thesis and an experimental apparatus consisting of an airtable and two robots built on CubeSat frames sitting on top of pucks so that they can float in the minimal friction environment generated by the airtable.

3.7.1 Experimental Apparatus

The hardware and the majority of the software for the experimental apparatus was designed, built, and implemented by Dr. Marin Kobilarov, a KISS‡ post-doctoral fellow at Caltech. As shown in Figure 3.91, the apparatus consists of an airtable, two mobile robots, a desktop computer, and an overhead camera.

In addition to the electromagnets, each robot has a set of six thrust vectored fans which can be

*The exact specifications of this system are currently under consideration.
†The specifications of the magnetic actuation system are also still under consideration.
‡KISS is the Keck Institute for Space Studies.

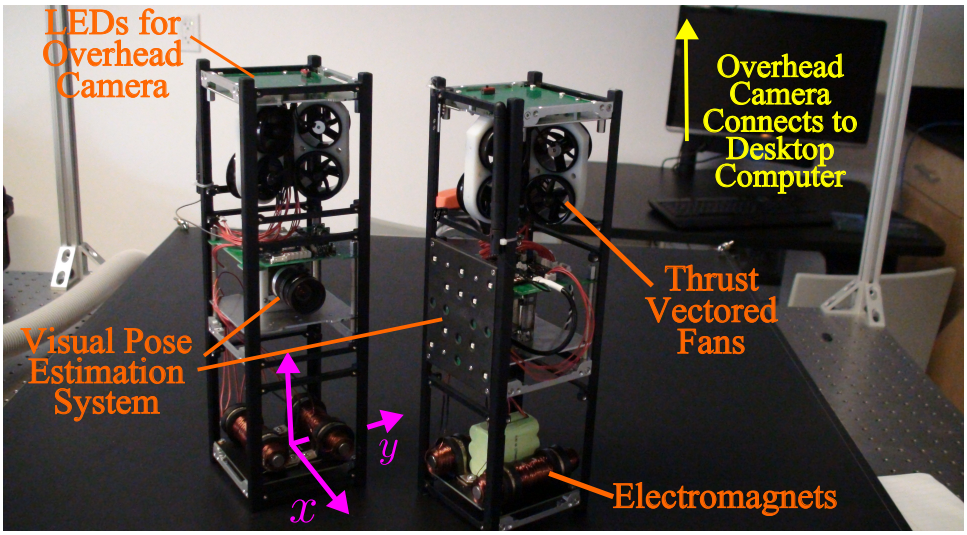


Figure 3.91: The essential components of the experimental apparatus consist of two robots sitting on the air table, an overhead camera used to track the position of the robots based on LEDs attached to the top of each robot, and a desktop which can be used for any processing not done onboard the robots, i.e. either the onboard visual pose estimation system or the overhead tracking can be used to provide feedback. Electromagnets are also shown in their current configuration. Pucks are not shown.

used–in conjunction with state feedback–to fully control the state of each robot. The fans were not used in conjunction with the electromagnets, but rather to verify control algorithms and facilitate disturbance quantification, as discussed in Appendix B. It should also be noted that the electromagnets shown in Figure 3.91 were not present for all of the experiments. In the initial stages of the docking port design and validation, a single permanent magnet was centered on the front of each spacecraft to provide a driving force for uncontrolled docking maneuvers.

In the following numerical experiments and optimization, the DCR-DCM approach was utilized with non-convex bodies decomposed into their convex parts for collision detection and momentum updates.

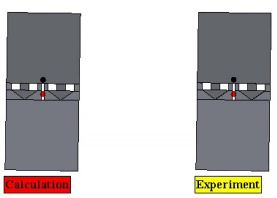
Permanent Magnets. The motion of the satellite is driven by permanent magnets either centered on the front of each spacecraft for the validation calculations, or embedded into the base of the cups and tips of the cones for the optimal port placement calculations. In the first case, relatively strong Aluminum-Nickel-Cobalt (AlNiCo) magnets were used, with a diameter of 0.95cm and a thickness of 0.25cm. The max pull was rated at $\sim$ 0.5lbs. In the latter case, 0.5cm diameter magnets were used with a max pull of 0.2lbs. Due to the relatively small size of the magnets and compared to the length scales in the model, each magnet was introduced by modeling two oppositely charged Coulombic charges separated by the thickness of the magnet. An effective parameter K was determined empirically by tuning the time scale of the simulation to match the experiment when

the CubeSat robots were separated by 15cm with their docking ports aligned.

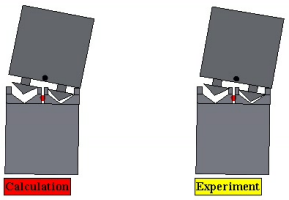
3.7.2 Kinematic Docking Port Design

The docking port designs described in this section were developed by students enrolled in the Ae105 (Aerospace Engeineering) course. The initial design was used to validate the use of the forward dynamics code developed in this thesis as a reasonable representation of the dynamics and kinematics of unactuated docking. The port design was overhauled by the following course; in order to avoid overconstraining the system and possible spalling and cold welding in a real orbital environment, a kinematic mount was introduced into the base of an improved cup-cone design. Using this new geometry, a series of simulations was used to determine the optimal placement of the docking ports to maximize the likelihood of successful docking driven by electromagnets using a representative 2d geometry.

Validation. A series of permanent-magnet driven tests were conducted on the air table with the docking port cup-and-cone geometry shown in Figure 3.92. The initial conditions from these results were fed into a 2D forward dynamics simulation designed mirror the essential dynamics of the dynamics on the air table, with an eye on verifying that the simulation was capable of capturing key behaviors that led to either successful or failed docking for the given port geometry. Specifically, this preliminary port design exhibited serious issues with jamming due to the tight fit between the cups and cones, as well as the interaction of the large cone angle with the cylindrical portion of the cups. As can be seen in Figures 3.92, this behavior is successfully captured by the dynamics simulation. Figure 3.93 compares all results of experimental and calculated trajectories.



(a) Successful docking in both experiment and simulation.



(b) Unsuccessful docking due to jamming in both experiment and simulation.

Figure 3.92: Docking port geometry used to validate the fidelity of the simulation.

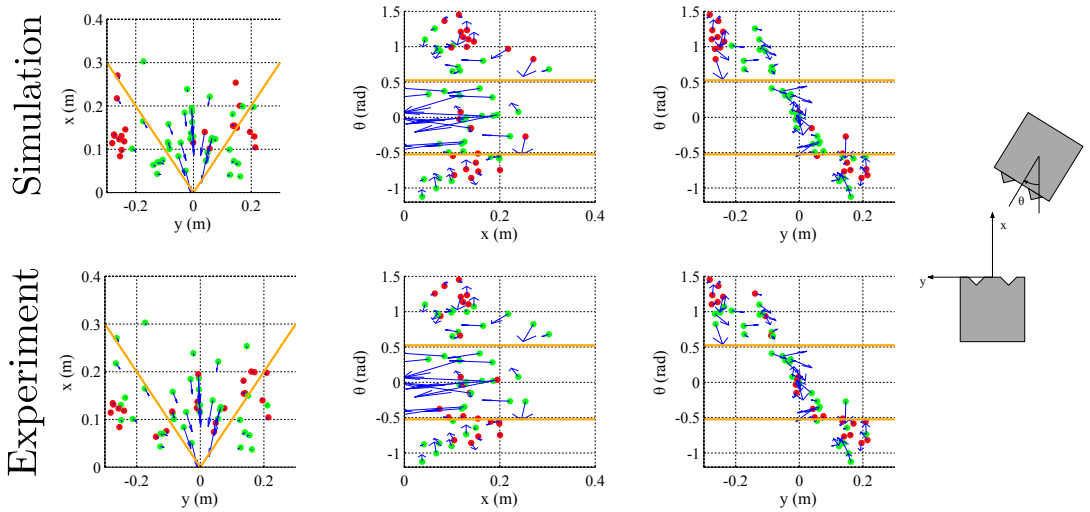
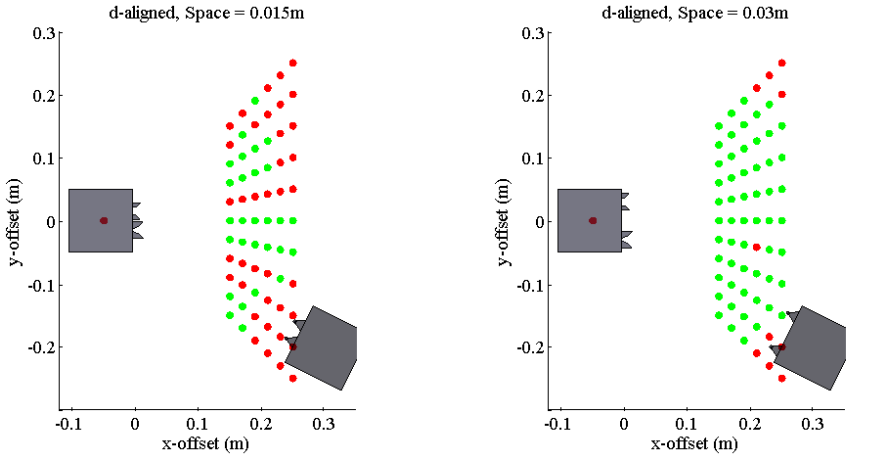


Figure 3.93: Comparison between end results (docking success in green or failure in red) of all experimental and calculated trajectories. The initial configurations are represented by red or green dots, with velocity vectors sketched as blue arrows. The simulation successfully captures the end results in the majority of cases, with discrepancies coming from the un-modeled disturbance field and other uncertainties in the experiment.

Optimal Port Placement. In additioned to the afforementioned issues related to jamming in the initial designs and with input from mechanisms experts from the Jet Propulsion Laboratory, a new docking port deisgn consisting of a kinematic mount embedded in a cup-and-cone geometry was implemented. The primary goals of this design were to avoid the jamming issues, reduce the likelihood of cold-welding by reducing the contact area between the cups and cones, and also to avoid overconstraining the system in the docked configuration to facilitate better alignment of the mirrors. In addition, small permenent magnets were embedded in the base of each cup and the tip of each cone in a preliminary test of a latching system.

Having redesigned the port geometry, the port placement in terms of the center-to-center spacing of the ports was optimized by direct calculation over a series of initial conditions as shown in Figures 3.94 and 3.95, which were driven by the embedded permanent magnets (i.e. a worst-case scenario if the electromagnets were to go offline). Empirical results from the Ae105 working group–which also made use of the electromagents shown in Figure 3.91–showed that offsetting the docking ports in the (unmodeled) $z-$direction docking further reduced docking failures due to misalignment. The final 3D design is shown in Figure 3.96.led) $z-$direction docking further reduced docking failures due to misalignment. The final 3d design is shown in Figure 3.96.



(a) Ports offset from center by 1.5cm

(b) Ports offset from center by 3.0cm

Figure 3.94: Successful or unsuccessful docking for improved docking port design. All initial velocities are zero, and the male spacecraft is aligned with a vector connecting the centers of the craft.

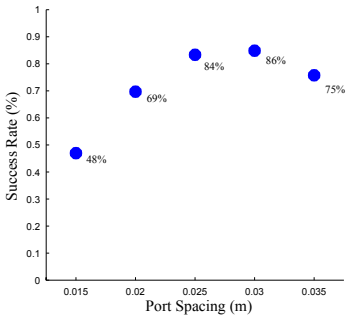


Figure 3.95: Docking success rate as a function of port spacing.

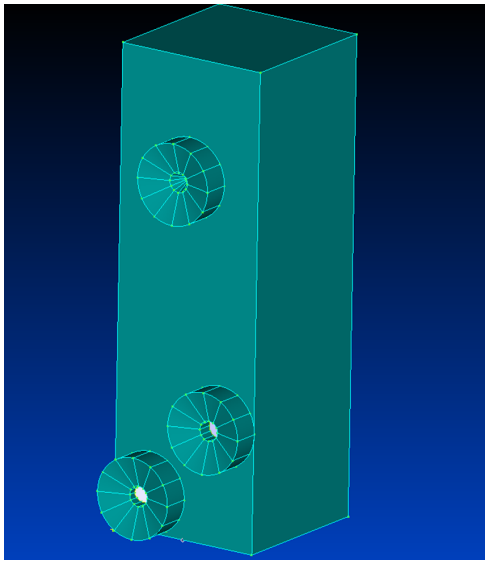


Figure 3.96: Final docking port configuration with ports slightly offset in the out-of-plane direction.

3.8 Conclusion

This chapter introduces three approaches to simulating dynamics with collisions emphasizing constrained rigid bodies, as summarized in Table 3.1, all of which employ the SSH LP to detect and describe collisions. In the most straightforward approach, a parameter-based contact potential is introduced. Given the simplicity of the approximation, the potential-based approach yields reasonable results, and can be tuned to increase or decrease the accuracy of the dynamics, with a corresponding decrease or increase in the efficiency of the integrator, but the necessity to allow overlap means that this algorithm is never completely accurate. However, this approach remains useful if high accuracy is not needed because of its good qualitative and reasonable quantitative results.

The exact approach is so named because it exactly solves the discrete impact equations, and therefore exactly preserves any conserved quantities in the system through the contact. However, this approach is hindered by the need to precisely resolve the order of collisions, which can lead to prohibitively small time intervals between collisions. In addition, if collisions cannot be decoupled (i.e. due to potentials which depend on the relative configuration of all bodies in the system) the need to solve the discrete impact equations for the full system of bodies significantly augments the expense of the algorithm.

The DCR-DCM approach represents an efficient approximation to the exact approach. Here, the use of a CPP operation to resolve an admissible contact configuration along with the assumption that all collisions occur simultaneously at the end of each time step circumvent the main difficulties of the exact approach, and allow for relatively large time steps to be taken. Multiple collisions are dealt with seamlessly, and all (small) non-physical changes to conserved quantities are due to the CPP operation, rendering this method a so-called conserving approximation. The DCR-DMC algorithm yields good quantitative and qualitative comparisons to the exact approach shown in Section 3.6. Furthermore, this approach leads to a straightforward way to introduce inelastic and even frictional collisions, and shows good stability in maintaining a small oscillation about a minimal energy configuration (i.e., a static solution) in dissipative systems.

4 Control Problems

4.1 Introduction and Organization

The preceding chapters have been devoted to the development of robust collision detection for non-smooth bodies and the development of Newmark and constrained variational integrators which account for collisions between such bodies. The development and use of structure preserving integrators (neglecting contact) akin to those developed in Chapter 3 in the formulation of optimal control problems–i.e., discrete mechanics and optimal control for constrained systems (DMOCC)–has been developed in detail in previous works (c. f. [52]), and will be outlined in Section 4.2. However, in many problems of interest collisions need to be specifically avoided. As will be shown in Section 4.3, this amounts to the inclusion of path constraints in the optimal control problem. For non-smooth bodies, this technique is specifically enabled by the use of the SSHLP algorithm. In addition to enabling optimal control maneuvers which avoid collisions between non-smooth bodies, the exactness of the SSH LP allow it to be used to design maneuvers which, e.g., move the system from one specified contact configuration to another and also avoid collisions between non-convex bodies during assembly processes which involve 'tight' maneuvering.

In other problems, for instance in hybrid systems in which some bodies might in the system be underactuated or have no actuation at all, collisions cannot be avoided. Furthermore, collisions are actually required to execute docking maneuvers. It then becomes necessary to explicitly *plan* for collisions as part of the maneuver by including them in the formulation of the optimal control problem. As such, the inclusion of both elastic and inelastic collisions in DMOCC problems is developed and discussed in Section 4.4. The new method makes use of the momentum reflection technique described in Section 3.5.4 at a series of pre-seeded time nodes, determined by an initial guess generated by a number of means, e.g. inverse dynamics, forward dynamics using a potential based approach, or forward dynamics using a fully conservative approach or the DCR-DMC algorithm. In this way, an exact solution to the dynamics problem is found, because the contact constraint that $g(\boldsymbol{q}_c) = 0$ is exactly satisfied at the solution to the optimal control problem.

In the final part of this chapter, Section 4.5.6, the problem of designing a robust docking system for the AAReST mission is treated. A robust feedback controller for docking maneuvers is developed. It is shown that for fully act uated systems, this controller is globally asymptotically stable to the desired docked configuration. The methods developed in the previous section are used to test the robustness of this controller when collisions are present in the dynamics.

4.2 Constrained Discrete Mechanics and Optimal Control (DMOCC)

A general optimal control problem consists of the minimization of a generic cost functional (e. g. control effort, total time, etc.) subject to a set of nonlinear constraints including the equations of motion, plus any additional constraints (e.g. boundary conditions, path constraints, total time, actuation, etc). In this section, the groundwork laid by Leyendecker et al. [51], which makes use of Leyendecker et al. [52] and Marsden and Ratiu [57], will be outlined in order to extend the use of the discrete mechanics and optimal control methodology to problems involving non-smooth collision avoidance and planned contact interactions. Before proceeding, it should be noted that the method described here and in the following sections falls into a category of problems in which *local* optima are found by direct optimization. Furthermore, the use of constrained mechanics to describe the optimal control problem is particularly useful because the specification of multibody interactions, actuator locations, and constraints can be readily specified in the redundant director formulation (c. f. [10, 49, 51]). This formulation also circumvents known issues with singularities in the parametrization of rotations (c. f. Bachau and Trainelli [7], Spurrier [80]).

4.2.1 Continuous Optimization Problem

The setting for the constrained continuous optimization problem is analogous to the constrained continuous forward dynamics problem described in Section 3.4.1, and revisited here. In particular, an $n-$dimensional mechanical system with time-dependent state $(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t))$, with $\boldsymbol{q} \in \mathcal{Q} \subseteq \mathbb{R}^n$ and $\dot{\boldsymbol{q}} \in T_{\boldsymbol{q}}\mathcal{Q}$, with the time $t \in [0, T] \subset \mathbb{R}$ and index $K \in \mathbb{N}$ is considered. For constrained dynamics, the state is required to evolve only on the $(n-m)-$dimensional submanifold $\mathcal{C} = \{\boldsymbol{q} \in \mathcal{Q} | \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}\}$ along with the tangent bundle $T\mathcal{C} = \{(\boldsymbol{q}, \dot{\boldsymbol{q}}) \in T_{\boldsymbol{q}}\mathcal{Q} | \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}, \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}\}$, for which $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^m$ are a set of holonomic, scleronomic constraints and $\boldsymbol{G}(\boldsymbol{q})$ is the Jacobian of the constraint vector $\boldsymbol{g}$. In the analog to (3.54), the dynamics of the forced system follow from a constrained version of the Lagrange-d'Alembert principle for a force field $\boldsymbol{f} : \mathbb{R}^{n-m} \times T\mathcal{Q} \to T^*\mathcal{Q}$ which states that

$$\delta \int_0^{t_K} \left(L(\boldsymbol{q}, \dot{\boldsymbol{q}}) - \boldsymbol{g}^T \boldsymbol{\lambda} \right) dt + \int_0^{t_k} \boldsymbol{f} \cdot \delta \boldsymbol{q} dt = 0 \tag{4.1}$$

must hold for all $\delta \boldsymbol{q} \in T\mathcal{Q}$ and $\delta \boldsymbol{\lambda} \in \mathbb{R}^m$, with initial and final conditions specified so that $\delta \boldsymbol{q}(0) = \delta \boldsymbol{q}(K) = \boldsymbol{0}$. The second integral in (4.1) represents the work done by the external forces on admissible virtual displacements $\delta \boldsymbol{q}$. The Lagrangian, $L(\boldsymbol{q}, \dot{\boldsymbol{q}})$, is given in (3.53), and is simply the kinetic energy minus the potential energy in the system. Following the development in (3.54) and neglecting any contact interactions, the resulting equations of motion are given by

$$\frac{\partial L}{\partial \boldsymbol{q}} - \frac{d}{dt} \frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \boldsymbol{G}^T(\boldsymbol{q}) \boldsymbol{\lambda} + \boldsymbol{f} = \boldsymbol{0} \tag{4.2a}$$

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}. \tag{4.2b}$$

Again, in the analog to the unforced system, the equations of motion can be reduced to a minimal set of equations via the null space method, in which the first set of equations in (4.2) are pre-multiplied by the transposed of a matrix $\boldsymbol{P}(\boldsymbol{q}) \in \mathbb{R}^{n \times (n-m)}$, which has columns orthogonal to the basis of $\boldsymbol{G}(\boldsymbol{q})$, that is, range$(\boldsymbol{P}(\boldsymbol{q}))$ = null$(\boldsymbol{G}(\boldsymbol{q})) = T_{\boldsymbol{q}}\mathcal{C}$. In addition, a local parametrization of the constraint manifold using a minimal set of variables $u \in \mathcal{U} \subseteq \mathbb{R}^{n-m}$, $\boldsymbol{F} : \mathcal{U} \to C$ may be employed to eliminate the second set of equations from (4.2). Furthermore, the forces $\boldsymbol{f} \in T^*_{\boldsymbol{q}}\mathcal{Q}$ can be parametrized in terms of a minimal set of generalized coordinates $\boldsymbol{\tau}(t) \in T^*\mathcal{U} \subseteq R^{n-m}$. The generalized forces can be calculated from the redundant forces as $\boldsymbol{\tau} = (\frac{\partial \boldsymbol{F}}{\partial \boldsymbol{u}})^T \boldsymbol{f}$, and the redundant forces can be determined from the generalized forces using the mapping $\boldsymbol{B}^T : T^*\mathcal{U} \to T^*\mathcal{Q}$ via $\boldsymbol{f} = \boldsymbol{B}(\boldsymbol{q})^T \boldsymbol{\tau}$ (c.f. [31, 52]). In a general optimal control problem, path constraints of the form $\boldsymbol{r}(\boldsymbol{u}, \dot{\boldsymbol{u}}, \boldsymbol{\tau}) \leq \boldsymbol{0}$ may also be introduced.

To formulate the objective functional, a mapping $B : T\mathcal{C} \times T^*_{\boldsymbol{q}}\mathcal{Q} \to \mathbb{R}$ is used as a continuous cost function. The objective functional, $J(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{f})$, is then given by

$$J(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{f}) = \int_0^{t_K} B(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{f}) dt, \tag{4.3}$$

with perhaps the most common example of a cost function being the total control effort, i. e. , $B = \frac{1}{2} \boldsymbol{f}^T \boldsymbol{f}$.

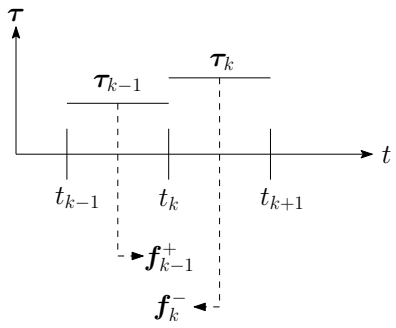


Figure 4.1: Relationship between generalized forces and redundant forces.

4.2.2 Discrete Optimization Problem

In the analog to the development of the constrained continuous optimal control problem in the previous section, the constrained discrete optimal control problem relies on a *discrete* constrained version of the Lagrange-D'Alembert principle to describe the forced discrete equations of motion. Recall that in the discrete case, the path is now composed of a discrete set of K configurations, $\{\boldsymbol{q}_k\}_{k=0}^K$ with $\boldsymbol{q}_k \in \mathcal{Q}$ at times $\{t_k\}_{k=0}^K$, and the continuous path $\boldsymbol{q} : [0, T] \to \mathcal{Q}$ is replaced by the discrete path $\boldsymbol{q}_d : \{t_k\}_{k=0}^K \to \mathcal{Q}$ with $K \in \mathbb{N}$. The virtual work term in Equation (4.1) in one time step is approximated in an analogous way to the constraint force term following

$$\boldsymbol{f}_k^- \cdot \delta\boldsymbol{q}_k + \boldsymbol{f}_k^+ \cdot \delta\boldsymbol{q}_{k+1} = \int_{t_k}^{t_{k+1}} \boldsymbol{f} \cdot \delta\boldsymbol{q} dt, \tag{4.4}$$

with

$$\begin{aligned} \boldsymbol{f}_k^- &= \frac{h}{2}\boldsymbol{B}^T(\boldsymbol{q}_k)\boldsymbol{\tau}_k \\ \boldsymbol{f}_{k-1}^+ &= \frac{h}{2}\boldsymbol{B}^T(\boldsymbol{q}_k)\boldsymbol{\tau}_{k-1}. \end{aligned} \tag{4.5}$$

In this description (see Figure 4.1), $\boldsymbol{f}_{k-1}^+, \boldsymbol{f}_k^- \in T_{\boldsymbol{q}_k}^*\mathcal{Q}$, represent the effect of the generalized forces, $\boldsymbol{\tau}_{k-1}, \boldsymbol{\tau}_k \in T^*\mathcal{U}$, acting in the intervals $[t_{k-1}, t_k]$ and $[t_k, t_{k+1}]$, respectively, on the configuration $\boldsymbol{q}_k$, with constant time step h. The constrained Lagrange-D'Alembert principle for the discrete system is requires that

$$\delta \sum_{k=0}^{K-1} \overline{L}_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{\lambda}_k, \boldsymbol{\lambda}_{k+1}, h) + \sum_{k=0}^{K-1} \left(\boldsymbol{f}_k^- \cdot \delta\boldsymbol{q}_k + \boldsymbol{f}_k^+ \cdot \delta\boldsymbol{q}_{k+1}\right) = \boldsymbol{0} \tag{4.6}$$

for all variations $\{\delta \boldsymbol{q}_k\}_{k=0}^K$ and $\{\delta \boldsymbol{\lambda}_k\}_{k=0}^K$, with $\overline{L}_d$ given in (3.7) and $\delta \boldsymbol{q}_0 = \delta \boldsymbol{q}_K = \boldsymbol{0}$. The constrained forced discrete Euler-Lagrange equations are then given by

$$D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, h) + D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k, h) - h\boldsymbol{G}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k + \boldsymbol{f}_k^- + \boldsymbol{f}_{k+1}^+ = \boldsymbol{0}, \tag{4.7a}$$

$$\boldsymbol{g}(\boldsymbol{q}_{k+1}) = \boldsymbol{0}. \tag{4.7b}$$

Again, (4.7) can be further reduced via premultiplcation by a discrete null space matrix, and a local nodal reparametrization in terms of discrete generalized coordinates, $\boldsymbol{u}_k \in \mathcal{U} \subseteq \mathbb{R}^{n-m}$. In particular, the discrete configurations $\boldsymbol{q}_k$ along the discrete path $\boldsymbol{q}_d$ are related by $\boldsymbol{q}_k = \boldsymbol{F}_d(\boldsymbol{u}_k, \boldsymbol{q}_{k-1})$. The mapping $\boldsymbol{F}_d : \mathcal{U} \times \mathcal{Q} \to \mathcal{C}$ depends on the nature of the system (i.e., whether any joints are present as discussed in Section 3.2), and is given for free rigid bodies in (3.24). The resulting scheme, with $\boldsymbol{g}(\boldsymbol{F}_d(\boldsymbol{u}_{k+1}, \boldsymbol{q}_k)) = \boldsymbol{0}$ is equivalent to the conservative scheme given in Equation 3.13 with the addition of the (now projected) discrete forces in (4.7)

$$\boldsymbol{P}^T(\boldsymbol{q}_k)\left[D_1 L_d(\boldsymbol{q}_k, \boldsymbol{F}_d(\boldsymbol{u}_{k+1}, \boldsymbol{q}_k), h) + D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k, h) + \boldsymbol{f}_k^- + \boldsymbol{f}_{k+1}^+\right] = \boldsymbol{0}. \tag{4.8}$$

These techniques to reduce the problem to its minimal form (i.e., minimal number of equations and variables) are advantageous for optimal control problems, because the number of variables and constraints are likewise reduced to the minimum possible number.

Forced $(\boldsymbol{q}, \boldsymbol{p})$ Scheme. A number of *forced* discrete Legendre transforms, analogous to those defined for the conservative constrained system in Section 3.2 can be defined and are essential to the specification of consistent boundary conditions. For instance, the transform yielding redundant quantities is defined as $\mathbb{F}^{cf} L_d{:}Q \times Q \to T^*Q$

$$\begin{aligned}
&\mathbb{F}^{cf^-} : (\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) \mapsto (\boldsymbol{q}_k, \boldsymbol{p}_k^-) \\
&\boldsymbol{p}_k^- = -D_1 L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, h) + \frac{h}{2}\boldsymbol{G}^T(\boldsymbol{q}_k) \cdot \boldsymbol{\lambda}_k - \boldsymbol{f}_k^- \\
&\mathbb{F}^{cf^+} : (\boldsymbol{q}_{k-1}, \boldsymbol{q}_k) \mapsto (\boldsymbol{q}_k, \boldsymbol{p}_k^+) \\
&\boldsymbol{p}_k^+ = D_2 L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k, h) - \frac{h}{2}\boldsymbol{G}^T(\boldsymbol{q}_k) \cdot \boldsymbol{\lambda}_k + \boldsymbol{f}_{k-1}^+,
\end{aligned} \tag{4.9}$$

which can directly be used as a time stepping scheme by requiring the matching momenta and

solving for the new configuration via solving $\boldsymbol{p}_k^+(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k) = \boldsymbol{p}_k^-(\boldsymbol{q}_k, \boldsymbol{q}_{k+1})$ for $\boldsymbol{q}_{k+1}$, finding $\boldsymbol{p}_{k+1}^+$ using $\boldsymbol{q}_{k+1}$, and so forth. Both the projected and reduced version of the forced discrete Legendre transforms can also be analogously defined, in which the Lagrange multipliers are eliminated from the system. The reduced transform ${}^P\mathbb{F}^{cf}L_d{:}Q \times Q \to T^*\mathcal{U}$ is given by

$$\begin{aligned} {}^P\boldsymbol{p}_k^- &= \boldsymbol{P}^T(\boldsymbol{q}_k)\left[-D_1L_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}) - \boldsymbol{f}_k^-\right] \\ {}^P\boldsymbol{p}_k^+ &= \boldsymbol{P}^T(\boldsymbol{q}_k)\left[D_2L_d(\boldsymbol{q}_{k-1}, \boldsymbol{q}_k) + \boldsymbol{f}_{k-1}^+\right], \end{aligned} \tag{4.10}$$

which in combination with the parametrization $\boldsymbol{F}_d$, results in a minimal description of the forced discrete constrained system.

Angular Momentum. By using the forced discrete Legendre transform, (4.9) the angular momentum for the forced discrete constrained system is given by

$$\boldsymbol{L} = \boldsymbol{\varphi} \times \boldsymbol{p}_\varphi + \boldsymbol{d}_I \times \boldsymbol{p}_I.$$

The angular momentum in the forced system changes exactly according to the applied forces and any potentials present which lack symmetry under $SO(3)$, rendering the scheme momentum-consistent. For systems with either no potentials or a symmetric potential, the change in angular momentum is given by

$$\boldsymbol{L}_{k+1} - \boldsymbol{L}_k = \boldsymbol{\varphi}_{k+1} \times \boldsymbol{f}_{\varphi_k}^+ + \boldsymbol{d}_{I\,k+1} \times \boldsymbol{f}_{I\,k}^+ + \boldsymbol{\varphi}_k \times \boldsymbol{f}_{\varphi_k}^- + \boldsymbol{d}_{I\,k} \times \boldsymbol{f}_{I\,k}^-.$$

Depending on the form of the mapping $\boldsymbol{B}^T(\boldsymbol{q}_k)$ used to determine $\boldsymbol{f}_k = \boldsymbol{B}^T(\boldsymbol{q}_k)\boldsymbol{\tau}_k$, this relationship can be further reduced. To simplify notation in future sections, we introduce the net torque for the constrained discrete system as

$$\boldsymbol{T}_k = \boldsymbol{\varphi}_{k+1} \times \boldsymbol{f}_{\varphi_k}^+ + \boldsymbol{d}_{I\,k+1} \times \boldsymbol{f}_{I\,k}^+ + \boldsymbol{\varphi}_k \times \boldsymbol{f}_{\varphi_k}^- + \boldsymbol{d}_{I\,k} \times \boldsymbol{f}_{I\,k}^-, \tag{4.11}$$

so that the torque applied over the interval beginning at t_k is $\boldsymbol{T}_k$. The difference

$$\boldsymbol{\Delta}_k = (\boldsymbol{L}_{k+1} - \boldsymbol{L}_k) - \boldsymbol{T}_k \tag{4.12}$$

will be used to verify the consistency of optimal control solutions.

Asymmetric Potentials. When asymmetric potentials are present in the system, there are non-vanishing terms associated with those potentials. The change in total angular momentum in this case is given by

$$\begin{aligned}\boldsymbol{L}_{k+1} - \boldsymbol{L}_k = \boldsymbol{\varphi}_{k+1} \times \left[\boldsymbol{f}^+_{\varphi_k} - \frac{h}{2} D_{\boldsymbol{\varphi}} V \left(\frac{\boldsymbol{q}_k + \boldsymbol{q}_{k+1}}{2}\right)\right] + \boldsymbol{d}_{I\,k+1} \times \left[\boldsymbol{f}^+_{I\,k} - \frac{h}{2} D_{\boldsymbol{d}_I} V \left(\frac{\boldsymbol{q}_k + \boldsymbol{q}_{k+1}}{2}\right)\right] \\ + \boldsymbol{\varphi}_k \times \left[\boldsymbol{f}^-_{\varphi_k} - \frac{h}{2} D_{\boldsymbol{\varphi}} V \left(\frac{\boldsymbol{q}_k + \boldsymbol{q}_{k+1}}{2}\right)\right] + \boldsymbol{d}_{I\,k} \times \left[\boldsymbol{f}^-_{I\,k} - \frac{h}{2} D_{\boldsymbol{d}_I} V \left(\frac{\boldsymbol{q}_k + \boldsymbol{q}_{k+1}}{2}\right)\right].\end{aligned}$$

This gives rise to a second term in the momentum consistency condition as

$$\begin{aligned}\boldsymbol{T}_V = -h \left(\frac{\boldsymbol{\varphi}_k + \boldsymbol{\varphi}_{k+1}}{2}\right) \times D_{\boldsymbol{\varphi}} V \left(\frac{\boldsymbol{q}_k + \boldsymbol{q}_{k+1}}{2}\right) \\ - h \left(\frac{\boldsymbol{d}_{I\,k} + \boldsymbol{d}_{I\,k+1}}{2}\right) \times D_{\boldsymbol{d}_I} V \left(\frac{\boldsymbol{q}_k + \boldsymbol{q}_{k+1}}{2}\right)\end{aligned} \tag{4.13}$$

so that $\boldsymbol{\Delta}_k$ is given by

$$\boldsymbol{\Delta}_k = (\boldsymbol{L}_{k+1} - \boldsymbol{L}_k) - (\boldsymbol{T}_k + \boldsymbol{T}_V).$$

Presence of Gravity. A gravitational force acting in the $\boldsymbol{e}_z$ direction exerts a net torque in the $\boldsymbol{e}_x$ and $\boldsymbol{e}_y$ directions. If gravity is present, then $\boldsymbol{\Delta}_k$ as defined in (4.12) should be augmented by the term

$$\boldsymbol{\Delta}_k = (\boldsymbol{L}_{k+1} - \boldsymbol{L}_k) - (\boldsymbol{T}_k + \boldsymbol{T}_{grav}) \tag{4.14}$$

with $\boldsymbol{T}_{grav}$ given by

$$\boldsymbol{T}_{grav} = \frac{h}{2}(\boldsymbol{\varphi}_{k+1} + \boldsymbol{\varphi}_k) \times (M_\varphi g\, \boldsymbol{e}_z)\,. \tag{4.15}$$

Boundary Conditions. The initial and final redundant configurations can be specified by their relationship to the redundant designated configurations $\boldsymbol{q}^0$ and $\boldsymbol{q}^K$ in terms of at most $n-m$ compatible independent constraints for each configuration by

$$\begin{aligned} \boldsymbol{D}_0(\boldsymbol{q}_0, \boldsymbol{q}^0) &= \boldsymbol{0} \\ \boldsymbol{D}_K(\boldsymbol{q}_K, \boldsymbol{q}^K) &= \boldsymbol{0}, \end{aligned} \tag{4.16}$$

in which $\boldsymbol{D}_0(\boldsymbol{q}_0, \boldsymbol{q}^0), \boldsymbol{D}_K(\boldsymbol{q}_K, \boldsymbol{q}^K) \in \mathbb{R}^{n-j}$ with $j \geq m$. This leaves flexibility to fix these configurations in terms of the absolute global coordinates or specify the relative locations of bodies involved in the maneuver. It is also possible to replace the equality constraints by at most $2(n-m)$ compatible independent inequality constraints, or some compatible combination of equality and inequality constraints.

The specification of initial and final velocity is a more subtle problem. When the dynamics for the forced discrete system are evolved on $\mathcal{Q} \times \mathcal{Q}$ as in (4.8), the velocity is not well defined, and therefore the specification of the initial velocity in the optimal control problem is not immediately clear. Furthermore, for a realistic problem, the initial and final specified velocities should be consistent with constraints on the velocity level, (i.e. $\boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{0}$), before being transformed to initial and final conditions on the momentum level. As described in [52], this is accomplished by the use of a standard continuous Legendre transform $\mathbb{F}L : T\mathcal{C} \to T^*\mathcal{C}$,

$$\mathbb{F}L : (\boldsymbol{q}, \dot{\boldsymbol{q}}) \mapsto (\boldsymbol{q}, \boldsymbol{p}) = (\boldsymbol{q}, D_2 L(\boldsymbol{q}, \dot{\boldsymbol{q}}))$$

which for a consistent velocity $\dot{\boldsymbol{q}} = \boldsymbol{P}(q)\boldsymbol{\nu}$, yields momenta which meet the momentum level constraints in (3.19). For completeness, the full specification of the initial and final velocity conditions, $\dot{\boldsymbol{q}}(t_0) = \dot{\boldsymbol{q}}^)$ and $\dot{\boldsymbol{q}}(t_K) = \dot{\boldsymbol{q}}^K$, transformed to conditions in the initial and final momenta, reads

$$\begin{aligned} \boldsymbol{P}^T(\boldsymbol{q}_0)\left[D_2 L(\boldsymbol{q}_0, \dot{\boldsymbol{q}}^0)\right] - {}^P\boldsymbol{p}_0^- &= \boldsymbol{0} \\ \boldsymbol{P}^T(\boldsymbol{q}_K)\left[D_2 L(\boldsymbol{q}_K, \dot{\boldsymbol{q}}^K)\right] - {}^P\boldsymbol{p}_K^+ &= \boldsymbol{0}. \end{aligned} \tag{4.17}$$

The conditions in (4.17) can be succinctly re-written with only a slight abuse of notation as $\boldsymbol{p}^{0,+} = D_2L(\boldsymbol{q}_0, \dot{\boldsymbol{q}}^0)$ and $\boldsymbol{p}^{K,-} = D_2L(\boldsymbol{q}_K, \dot{\boldsymbol{q}}^K)$.

Discrete Objective Function. To formulate the objective functional in the discrete case, the integral expression in (4.3) is approximated according to

$$B_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{f}_k, h) = hB\left(\frac{\boldsymbol{q}_k + \boldsymbol{q}_{k+1}}{2}, \frac{\boldsymbol{q}_{k+1} - \boldsymbol{q}_k}{h}, \boldsymbol{f}_k\right) \approx \int_{t_k}^{t_{k+1}} B(\boldsymbol{q}, \dot{\boldsymbol{q}}, \boldsymbol{f})dt, \tag{4.18}$$

so that the discrete objective function is given by

$$J_d(\boldsymbol{q}_d, \boldsymbol{f}_d) = \sum_{k=0}^{K-1} B_d(\boldsymbol{q}_k, \boldsymbol{q}_{k+1}, \boldsymbol{f}_k, h). \tag{4.19}$$

In the fully reduced formulation, which is most conducive to the constrained discrete optimal control problem, the objective functional can be expressed as

$$\bar{J}_d(\boldsymbol{u}_d, \boldsymbol{\tau}_d) = \sum_{k=0}^{K-1} \bar{B}_d(\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\tau}_k, h). \tag{4.20}$$

The Optimal Control Problem. The developments in this section can be effectively summarized by the formulation of the following optimal control problem for a constrained discrete mechanical system, in which some or all of the states $(\boldsymbol{q}^0, \dot{\boldsymbol{q}}^0)$ and $(\boldsymbol{q}^K, \dot{\boldsymbol{q}}^K)$ have been specified:

$$\begin{aligned}
&\min_{\boldsymbol{u}_d, \boldsymbol{\tau}_d} \sum_{k=0}^{K-1} \bar{B}_d(\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\tau}_k, h) \\
&\quad \text{subject to:} \\
&\quad\quad {}^P\boldsymbol{p}_k^+ - {}^P\boldsymbol{p}_k^- = \boldsymbol{0}, \\
&\quad\quad \boldsymbol{s}_0(\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{\tau}_0, \boldsymbol{q}^0, \dot{\boldsymbol{q}}^0) = \boldsymbol{0}, \\
&\quad\quad \boldsymbol{s}_K(\boldsymbol{u}_{K-1}, \boldsymbol{u}_K, \tau_{K-1}, \boldsymbol{q}^K, \dot{\boldsymbol{q}}^K) = \boldsymbol{0}, \\
&\quad\quad \boldsymbol{h}_q(\boldsymbol{q}_k, {}^P\boldsymbol{p}_k) \leq \boldsymbol{0}, \\
&\quad\quad \boldsymbol{h}_\tau(\boldsymbol{q}_k, \boldsymbol{\tau}_k) \leq \boldsymbol{0}, \\
&\quad\quad \text{for } k = 1 \ldots K-1.
\end{aligned} \tag{4.21}$$

In (4.21), the equations of motion for the forced discrete system are enforced in the first set of constraints, $\boldsymbol{s}_0$ and $\boldsymbol{s}_K$ are conditions on the initial and final states of the system, and $\boldsymbol{h}_q$ and $\boldsymbol{h}_\tau$ represent path constraints, which for convenience have been divided into constraints on the state and constraints on the generalized forces, respectively. While not noted explicitly, all functions of the redundant configurations $\boldsymbol{q}_d \in \mathcal{Q}$ are necessarily related by the parametrization $\boldsymbol{q}_{k+1} = \boldsymbol{F}(\boldsymbol{u}_{k+1}, \boldsymbol{q}_k)$, so that (4.21) is stated in terms of a minimal number of variables and constraints.

Accuracy and Efficiency. It has been shown by Ober-Blöbaum et al. [66] that, for *unconstrained* dynamics (DMOC), if the discrete Lagrangian and virtual work approximate the continuous expressions to order κ accuracy, then the optimal control problem is also of order κ. For the constrained scheme, if the order of approximation of the Lagrange multipliers is also κ, then the accuracy of the solution is expected to be of $\mathcal{O}(h^\kappa)$ since the resulting trajectories are equivalent to the use of DMOC in generalized coordinates. Accuracy can be improved by improving the discretization in $\mathcal{Q} \times \mathcal{Q}$ by decreasing the time step h. As has been noted, the formulation of the constrained optimal control problem using the reduced forced Legendre transform, along with a parametrization in terms of the minimal generalized coordinates results in a nonlinear optimization problem with the minimal set of variables and constraints. *Locally* optimal solutions–which have some dependence on the initial guess–can be found by recourse to either sequential quadratic programming (SQP) or interior point (IP) methods (c.f. Nocedal and Wright [65]).

This chapter's focus is the introduction and demonstration of two new methods for planning maneuvers with non-smooth bodies in which collisions must be planned for or avoided, so the problem of finding globally optimal solutions to the DMOCC problems under consideration is left for future work. However, for many problems of interest, the secondary goal of designing the motion is nearly as important as minimizing the objective functional. It will be shown that in addition to supplying a locally optimal feasible trajectory, the local optima constitute significant improvements over the initial guess.

4.3 Optimal Control with Collision Avoidance

Consider a system with one or more possibly non-convex rigid bodies under actuation, and which could also include one or more un-actuated bodies which restrict the admissible configurations along the discrete path $\boldsymbol{q}_d = \{\boldsymbol{q}_k\}_{k=0}^K$. Let the contact set I_c index all possible *potential* collisions between the convex bodies in the system, and denote $|I_c| = N_c$. At each configuration in $\boldsymbol{q}_d$ take $|I_{c,k}| = N_{c,k}$. In a simple system with two convex bodies, $N_{c,k} = 1$; for three convex bodies, $N_{c,k} = 3$, and so forth.

If non-convex bodies are present in the system, then each non-convex body should be subdivided into convex sub-bodies, and all possible contacts between all convex sub-bodies (excluding self-interaction for rigid bodies) constitute the contact set. It goes without saying that, in order to formulate a minimal set of constraints, there should be exactly one interpenetration function, $g_c(\boldsymbol{q})$, for each pair of convex bodies indexed by I_c; for non-convex bodies or sub-bodies, $g_c(\boldsymbol{q})$ is given by the SSH LP, developed in Chapter 2. Depending on the required accuracy of the non-interpenetration constraints, it is also possible to use the unilateral contact potential based on the SSH LP, $V_{\mathcal{A}}(\boldsymbol{q}_k)$ given in (3.38) as a single path constraint at each node.

With these preliminaries in hand, the collision avoidance problem amounts to introducing a set of path constraints at each time node in the form $g_c(\boldsymbol{q}_k) - \epsilon \leq 0$, $c = 1 \ldots N_c$, where $\epsilon \geq 0$ is the safety margin by which possible obstacles should be be avoided. If the contact potential is used, then a precise margin cannot be specified, however by requiring $V_{\mathcal{A}}(\boldsymbol{q}_k) \leq 0$, no overlap will be allowed. The path constraints at each configuration are then given by $\boldsymbol{h}_q(\boldsymbol{q}_k) \in \mathbb{R}^{N_{c,k}}$ if each constraint is considered separately or $\boldsymbol{h}_q(\boldsymbol{q}_k) \in \mathbb{R}$ if the unilateral potential is used are then the vector of obstacle avoidance constraints.

Initial Guess. Before moving on to several illustrative examples of this methodology for non-smooth and non-convex bodies, the generation of a feasible initial guess with respect to the path constraints is treated. This is particularly important because, due to the non-convexity of the constraints in the DMOCC problem, standard SQP methods can only guarantee local optimality. While a global optimal solution could be found by recourse to, e.g., an outer optimization loop using an evolutionary or stochastic method, the problem of globally optimal solutions will not be treated herein. However, standard infeasible start procedures may not be able to find a locally feasible solution. Thus, the initial guess does have an influence on the final (locally optimal) solution, but in a worst-case scenario, a poor initial guess could prevent the optimizer from converging to *any* feasible solution even if one or more local minimal do exist.

For the collision avoidance problem, the examples in this section make use of a *mostly* feasible initial guess in which the initial configuration conditions, dynamics, and configuration path constraints are met. However, the initial momentum, the final momentum and configuration conditions, and any bounds on the generalized forces might not be met.

Initial guesses are generated in one of two ways. In the first example, the initial guess for the generalized forces $\boldsymbol{\tau}_{d,ig} = \{\boldsymbol{\tau}_{k,ig}\}_{k=1}^{K-1}$ is determined empirically by a guess-and-check method, and the generalized coordinates $\boldsymbol{u}_{d,ig} = \{\boldsymbol{u}_{k,ig}\}_{k=1}^{K-1}$ taken from the resulting forward dynamics simulation. In the final two examples, which demonstrate the optimal assembly of a parlor puzzle and optimal

control of an articulated manipulator, generalized forces that lead to a feasible path with $\boldsymbol{h}_q(\boldsymbol{q}_k) \leq \boldsymbol{0}$ for all $\boldsymbol{q}_k \in \boldsymbol{q}_d$ are back-calculated from a prescribed feasible path, $\boldsymbol{q}_{d,ig} = \{\boldsymbol{q}_{k,ig}\}_{k=0}^{K}$. That is, a sub-optimal initial guess is formulated by first determining the discrete path, $\boldsymbol{q}_{d,ig}$, and then solving the inverse dynamics problem in (4.22) to determine the generalized forces, $\boldsymbol{\tau}_{d,ig} = \{\boldsymbol{\tau}_{k,ig}\}_{k=0}^{K-1}$, with $\boldsymbol{\tau}_{0,ig} = \boldsymbol{0} \in \mathbb{R}^{n-m}$. Finally, a forward dynamics problem is solved using $\boldsymbol{\tau}_{d,ig}$ with the consistent initial momentum given in (4.23) to verify the inverse dynamics solution and recover the generalized configuration variables for the incremental updates, $\boldsymbol{u}_{d,ig} = \{\boldsymbol{u}_{k,ig}\}_{k=0}^{K-1}$.

$$\begin{aligned}\frac{h}{2}\boldsymbol{P}^T(\boldsymbol{q}_{k,ig})\boldsymbol{B}^T(\boldsymbol{q}_{k,ig})\boldsymbol{\tau}_{k,0} = -\boldsymbol{P}^T(\boldsymbol{q}_{k,ig})\,[&D_1 L_d(\boldsymbol{q}_{k,ig}, \boldsymbol{q}_{k+1,ig}) \\ &+ D_2 L_d(\boldsymbol{q}_{k-1,ig}, \boldsymbol{q}_{k,ig}) + \boldsymbol{f}^+_{k-1,ig}\Big]\,, k = 1, \ldots K-1\end{aligned} \tag{4.22}$$

$$\begin{aligned}\boldsymbol{p}^+_{0,ig} &= \boldsymbol{M}\boldsymbol{P}(\boldsymbol{q}_{0,ig})\boldsymbol{\nu}_0, \;\; \text{with:} \\ \boldsymbol{\nu}_0 &= -\boldsymbol{M}^{-1}_{red}\boldsymbol{P}^T(\boldsymbol{q}_{0,ig}) D_1 L_d(\boldsymbol{q}_{0,ig}, \boldsymbol{q}_{1,ig})\end{aligned} \tag{4.23}$$

In general, this path is determined by joining together a series of constant velocity–in terms of the twist for each body–maneuvers. This strategy renders the determination of a collision-free path relatively straightforward, however the inverse dynamics problem leads to sharp variations in the generalized forces in order to stop momentum in one direction and start it in another.

4.3.1 Examples

The following examples illustrate the utility of path constraints based on the SSH LP for several types of maneuvers for non-smooth and non-convex bodies. In each case, the initial guess is a collision-free trajectory which is internally consistent, but may not meed the initial and final state constraints. In addition to attaining final and initial states, the solution to the optimal control problems show a significant improvement to the discrete cost function when compared to the initial guess. The momentum-consistency of the solution is also verified in each case.

Idealized Small Satellite Reconfiguration. This example is concerned with the reconfiguration of two free non-convex rigid bodies, as shown in Figure 4.2, which represent idealized small satellites. The density of the satellites is $\rho = 27^{\text{kg}}/_{m^3}$, and the side length of the base cube is $l = 0.1$m. The goal is to moved from the initial 'docked' state, which represents a contact configuration with zero

initial momentum, to an alternate docked state, with zero final momentum. Both the initial and final states are fully specified w.r.t. the reference coordinate system. The non-convex bodies being modeled are each decomposed into three convex sub-bodies, as indicated in Figure 4.2. In this example, each body is assumed to have full (6 DoF) actuation. As such, the generalized forces for each body are the force and torque on that body given by

$$\boldsymbol{\tau}^j = \begin{bmatrix} \boldsymbol{\tau}_\varphi \\ \boldsymbol{\tau}_T \end{bmatrix} \in \mathbb{R}^6,$$

and the configuration-dependent redundant forces can be recovered as

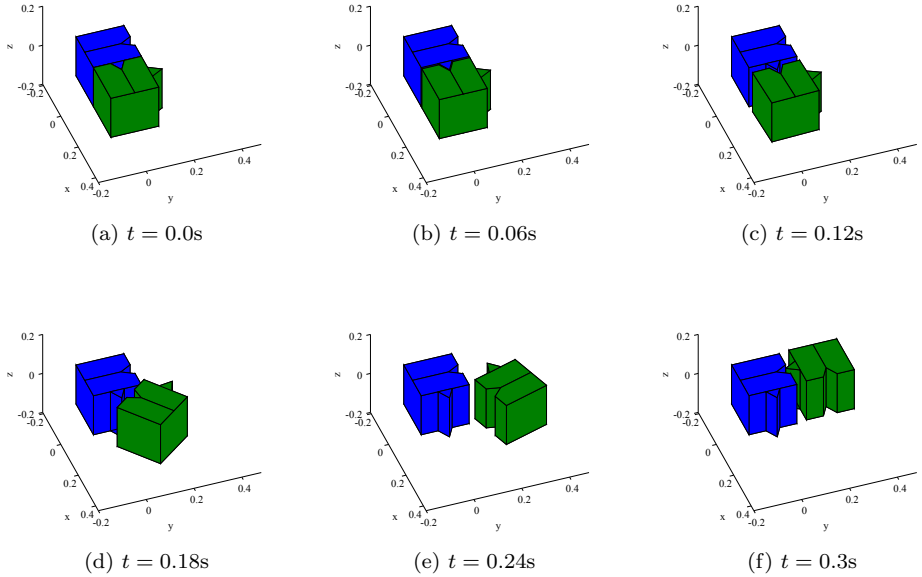
$$\begin{aligned} \boldsymbol{B}^T(\boldsymbol{q}^j) &= \boldsymbol{P}_{int}(\boldsymbol{q}^j) \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{2}\boldsymbol{I} \end{bmatrix} \boldsymbol{C}_{rb}(\boldsymbol{q}^j) \in \mathbb{R}^{12\times 6} \\ \text{with: } \boldsymbol{C}_{rb}(\boldsymbol{q}^j) &= \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \widehat{\boldsymbol{\varrho}}^{rb} & \boldsymbol{I} \end{bmatrix} \in \mathbb{R}^{6\times 6}. \end{aligned} \tag{4.24}$$

In general, $\boldsymbol{\varrho}^{rb} = \varrho_I^{rb}\boldsymbol{d}_I$ is the point at which $\boldsymbol{\tau}_\varphi$ are applied to the body. In the present example, $\boldsymbol{\varrho} = \boldsymbol{0}$, so that $\boldsymbol{\tau}_\varphi$ is applied at the center of mass.

The total time allotted for the maneuver, in which no bounds are placed on the control forces or torques, is $t_K = 0.3$s, and the time step $h = 0.01$s, so that $K = 30$ and there are $K + 1 = 31$ nodes in the discrete trajectory. The momentum-consistency of the solution is shown in Figure 4.4, illustrating the result that changes to the total angular momentum vector, $\boldsymbol{L}$, are exactly due to the applied control torques. Representative configurations in for the final solution are shown in Figure 4.5, which differ substantially from the ad-hoc initial guess. The control effort given by

$$\boldsymbol{J}_d = h \sum_{k=0}^{K-1} \boldsymbol{\tau}_k^T \boldsymbol{\tau}_k$$

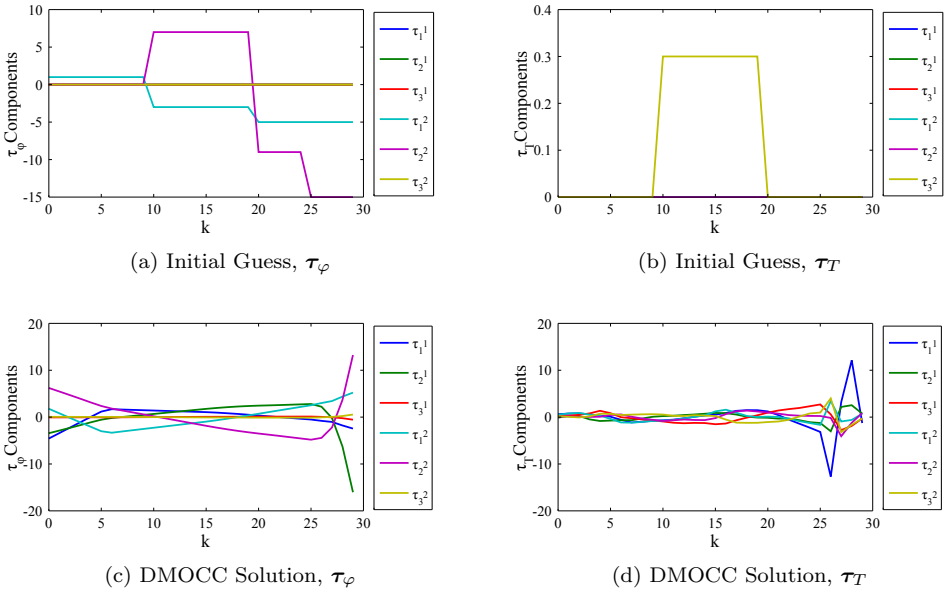
is reduced from $2.371e{+}1N \cdot s$ in the initial guess to $1.650e{+}1N \cdot s$ in the optimal solution, an improvement by a factor of about 1.5. A comparison between the generalized forces in the initial guess and final solution is shown in Figure 4.3.



(a) $t = 0.0$s (b) $t = 0.06$s (c) $t = 0.12$s

(d) $t = 0.18$s (e) $t = 0.24$s (f) $t = 0.3$s

Figure 4.2: Initial guess for satellite reconfiguration. Note that the final configuration condition is not met, and neither is the final momentum condition. Furthermore, the maneuver is realized by only moving one (green) satellite around the second stationary (blue) satellite.



(a) Initial Guess, $\boldsymbol{\tau}_\varphi$ (b) Initial Guess, $\boldsymbol{\tau}_T$

(c) DMOCC Solution, $\boldsymbol{\tau}_\varphi$ (d) DMOCC Solution, $\boldsymbol{\tau}_T$

Figure 4.3: Comparison of generalized control forces in the inverse dynamics based initial guess and the final optimal solution for the small satellite reconfiguration problem.

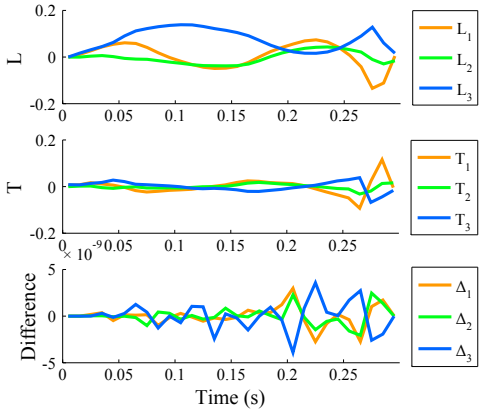
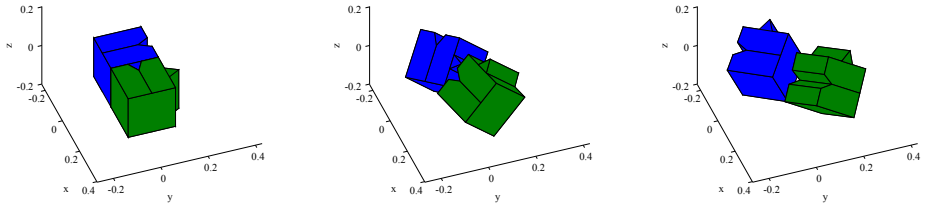
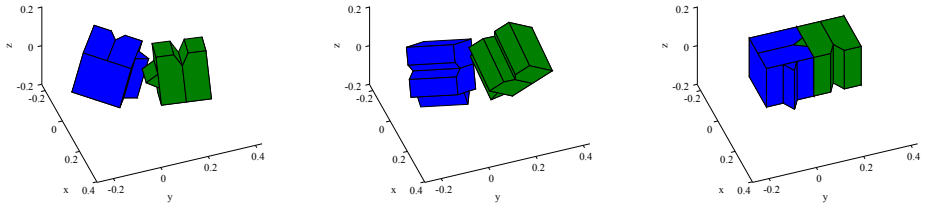


Figure 4.4: Momentum consistency of the optimal control solution for the small satellite reconfiguration problem. The tolerance on constraint violation used for the SQP solver in this problem is $1e-9$N·s and violations of the consistency condition are on the same order.



(a) $t = 0.0$s (b) $t = 0.06$s (c) $t = 0.12$s



(d) $t = 0.18$s (e) $t = 0.24$s (f) $t = 0.3$s

Figure 4.5: DMOCC solution with collision avoidance for the small satellite reconfiguration problem. Control effort is saved in the optimal solution by the application of a torque to both bodies, which reduces the overall effort needed for the reconfiguration.

Knot-Burr Puzzle Solution. As in the previous example, all bodies in this example are free (i.e., there are no joints), and all translational control forces are applied at the center of mass of each body. This example illustrates the optimization of a complicated assembly maneuver involving three non-convex bodies. It is inspired by the Knot-Burr parlor puzzle, shown in Figure 4.6. The initial guess was generated via solving the inverse dynamics problem to determine appropriate actuation to maintain a prescribed discrete trajectory, as shown in Figure 4.8. Figure 4.7 shows the quantitative and qualitative difference between the actuation from the inverse dynamics problem, and at the optimal solution, and Figure 4.10 shows that the solution is indeed momentum consistent. The initial configuration and zero initial momentum are fully specified in the reference coordinate system. Rather than being absolutely specified, the final configuration is constrained by the relative placement of each body, and the final relative momentum is likewise specified. Figure 4.9 shows several images of the assembly maneuver at the result of the optimal control problem. The uniform density of each puzzle piece is $700^{\text{kg}}/_{m^3}$, and the total time allowed for the maneuver is $t_K = 0.29$s at a time step of $h = 0.01$s, so that there are $K + 1 = 30$ nodes in $\boldsymbol{q}_d$. The control effort–calculated in the same way as in the previous example–for the initial guess is $8.490e{+}1$N·s. In the solution to the DMOCC problem, this is reduced by three orders of magnitude to $6.300e{-}2$N·s. This reduction is due to two factors. First, the impulsive nature of the trajectory used in the initial guess is replaced by smoother accelerations which require smaller magnitude control forces and torques. Second, there is more rotation of the pieces in the DMOCC solution, and rotational motion is cheaper than translational motion.

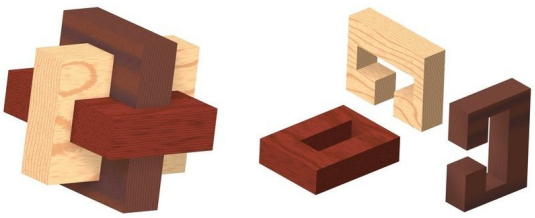
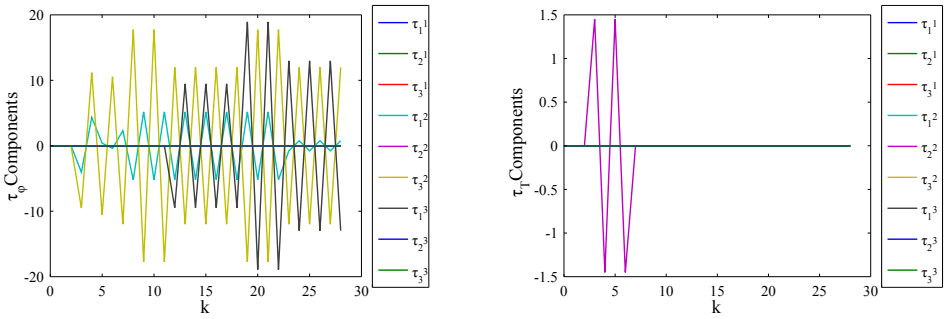


Figure 4.6: Knot-Burr puzzle assembly and parts (source: www.craftsmanspace.com).



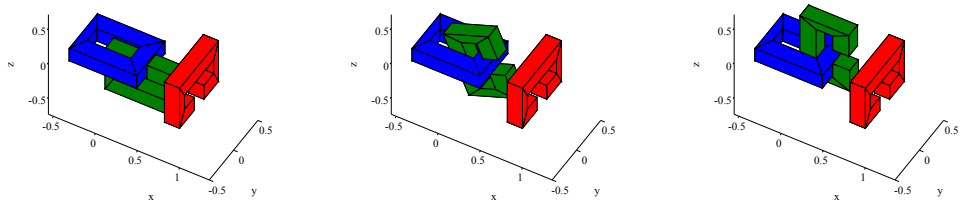
(a) Initial Guess, $\boldsymbol{\tau}_{\varphi}$

(b) Initial Guess, $\boldsymbol{\tau}_{T}$

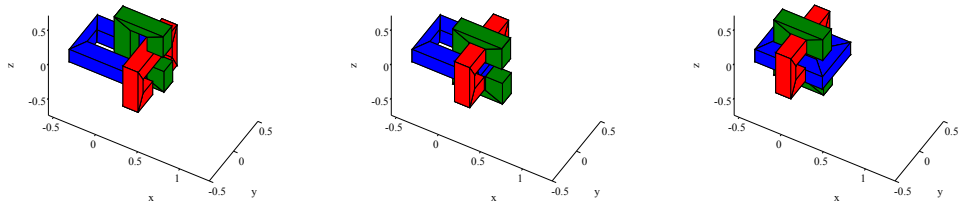
(c) DMOCC Solution, $\boldsymbol{\tau}_{\varphi}$

(d) DMOCC Solution, $\boldsymbol{\tau}_{T}$

Figure 4.7: Comparison of generalized control forces in the inverse dynamics based initial guess and the final optimal solution to the Knot-Burr puzzle assembly problem. The intuitive series of configurations used in the initial guess do not lead to an intuitive initial solution for the control. In fact, the impulsive nature of the initial motion–which is a series of constant velocity sub-trajectories–necessitates an impulsive and clearly sub-optimal set of initial control forces.

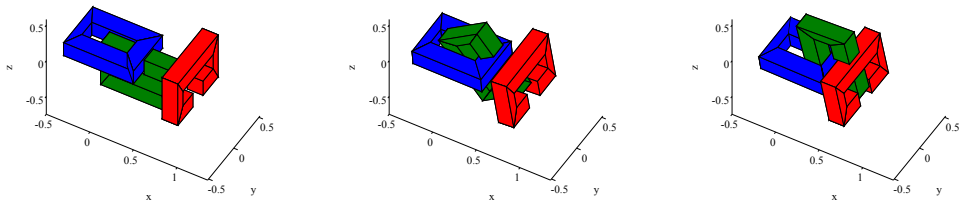


(a) $t = 0.0$s (b) $t = 0.06$s (c) $t = 0.12$s

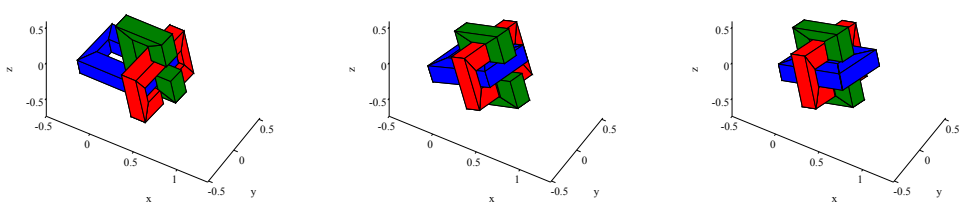


(d) $t = 0.18$s (e) $t = 0.24$s (f) $t = 0.29$s

Figure 4.8: Initial guess for assembly of the Knot-Burr puzzle. Note that the final configuration condition is met, however the final momentum condition is not. Furthermore, the maneuver is realized by only moving two (the red and green) pieces relative to the blue piece.



(a) $t = 0.0$s (b) $t = 0.06$s (c) $t = 0.12$s



(d) $t = 0.18$s (e) $t = 0.24$s (f) $t = 0.29$s

Figure 4.9: Results of the DMOCC problem for assembly of the Knot-Burr puzzle. The DMOCC solution yields a smooth maneuver in which actuation is applied to all puzzle pieces to effect the solution.

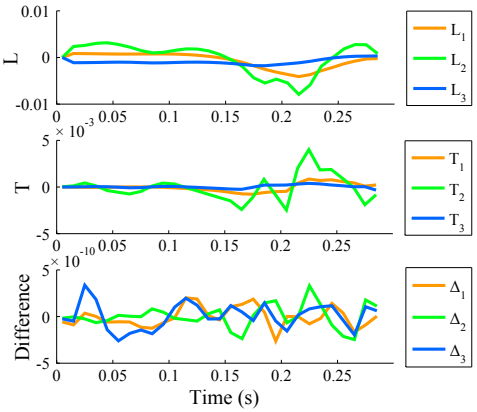


Figure 4.10: Momentum consistency of the optimal control solution to assembly of the Knot-Burr puzzle. The tolerance on constraint violation used for the SQP solver in this problem is $1e-9$N·s and violations of the consistency condition are on the same order.

Articulated Manipulator. This final example illustrates the optimal control of a kinematic chain in the presence of gravity. The goal of the so-called articulated manipulator is to move through the hole in the stationary block, and then form a hook around it, e. g. in preparation to pick up and move the block. Shown in Figure 4.12, the starting positon with the manipulator not aligned with the hole introduces an additional challenge into planning the maneuver. In addition, the final hooked configuration represents a contact configuration between the succcessive bodies, and overlap between the bodies must be avoided throughout the maneuver. A schematic of the manipulator, which consists of four bodies joined by three joints, is shown in Figure 4.11. The two revolute joints (R_1 and R_3) linking bodies 1 and 2, and 3 and 4, respectively, restrict relative rotations of the bodies to rotations about the axes $\boldsymbol{n}^1$ and $\boldsymbol{n}^3$ which are defined naturally by the shape of the bodies. Likewise, the position of the spherical joint (S) linking bodies 2 and 3 is also given naturally by the geometry of bodies 1 and 2.

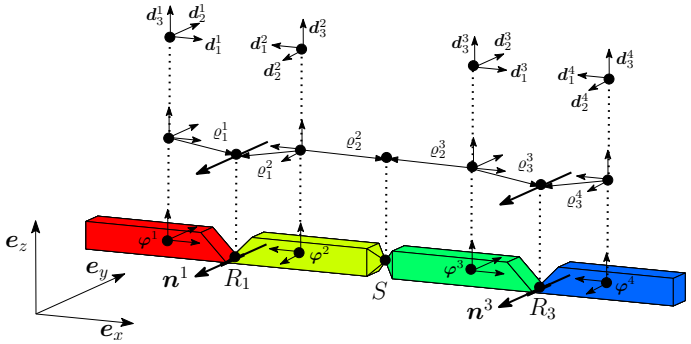


Figure 4.11: Schematic diagram of the assembled kinematic chain of four polyhedral bodies used to model an articulated manipulator. Joint locations and kinemttic constraints are induced naturally from the geometry of the bodies.

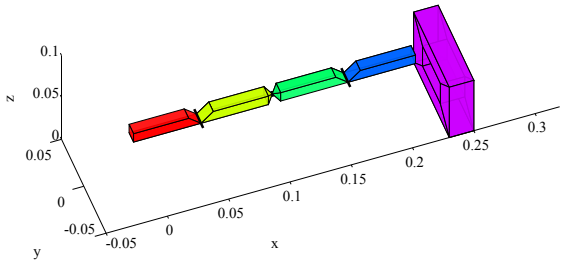


Figure 4.12: Initial configuration for the four-body articulated manipulator. Note that the tip of the manipulator (i.e., the fourth body in the chain) must move in the negative $\boldsymbol{e}_z$ and $\boldsymbol{e}_y$ directions to avoid hitting the stationary block as it approaches the hole.

In the final hooked configuration (see, e. g. , Figure 4.14f), the flat surfaces adjacent to the joints are required to be in contact, so that the relative orientation of each body w.r.t. its pair is specified. The

absolute orientation of the assembly, however, is given some leeway in that the center of mass of body three is required to be in the box defined by the hole of the stationary block. The final orientation of the manipulator is restricted by $\boldsymbol{d}_3^{3^T}\boldsymbol{e}_z \geq 0.8$, which means that the hook must be generally pointing 'up', but the upper surface of body 3 does not necessarily need to be parallel to any of the internal surfaces of the stationary block. In the DMOCC solution minimizing control effort, this results in contact between body 4 and the stationary block in the final configuration.

The generalized forces in this example, $\boldsymbol{\tau}$ are given by

$$\boldsymbol{\tau} = \begin{bmatrix}\boldsymbol{\tau}_\varphi^1 & \boldsymbol{\tau}_T^1 & \tau^{R_1} & \boldsymbol{\tau}^S & \tau^{R_3}\end{bmatrix}^T \in \mathbb{R}^{11},$$

in which $\boldsymbol{\tau}_\varphi^1$, $\boldsymbol{\tau}_T^1 \in \mathbb{R}^3$ are the forces and torques on the first body, τ^{R_1}, $\tau^{R_3} \in \mathbb{R}^1$ are joint torques applied in the revolute joints, and $\boldsymbol{\tau}^S \in \mathbb{R}^3$ are joint torques applied in the spherical joint.

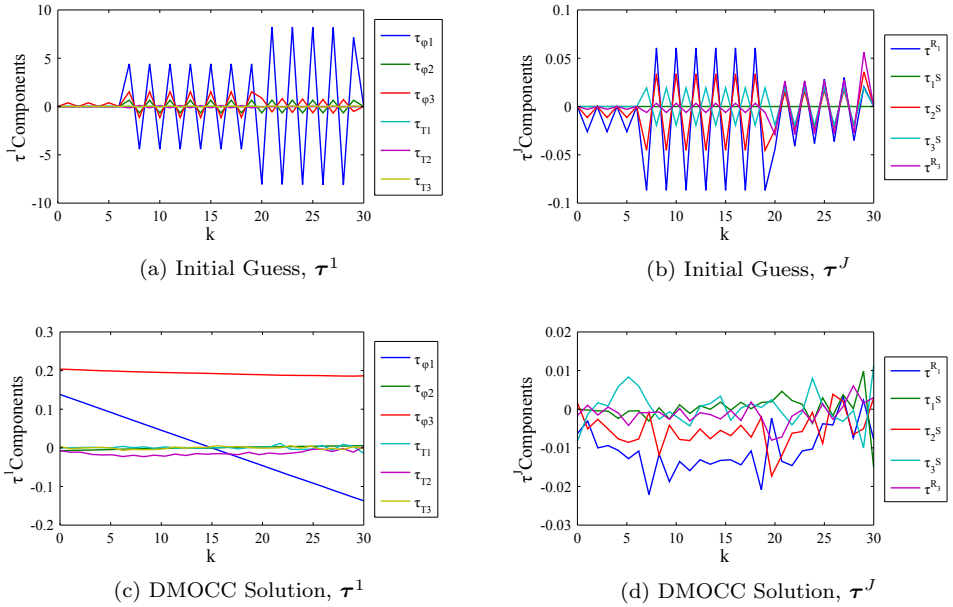
Recovering the redundant joint forces from the generalized joint forces bears additional discussion, which can be found in Appendix A. The general formulation, analogous to the statement for free rigid bodies in the previous two examples, is that the forcing on the chain due to $\boldsymbol{\tau} = \begin{bmatrix}\boldsymbol{\tau}^1 & \boldsymbol{\tau}^C\end{bmatrix}^T$ is

$$\begin{bmatrix}\boldsymbol{f}^1\\ \boldsymbol{f}^2\\ \boldsymbol{f}^3\\ \boldsymbol{f}^4\end{bmatrix} = \boldsymbol{B}^{C^T}(\boldsymbol{q})\boldsymbol{\tau},$$

in which the mapping $\boldsymbol{B}^{C^T}(\boldsymbol{q}) \in \mathbb{R}^{48\times 6+m_{ext}^C}$ has the same general form as (4.24), with $\boldsymbol{C}_{rb}$ replaced by $\boldsymbol{C}^C$.

The null space matrix for the complete chain is determined in an analogous way to the discussion in Section 3.2 by analysis of relationships between joint velocities, and is given in detail in Appendix A along with the explicit update for the chain.

With the preliminaries aside, we move on to the essential results of the minimal control effort reconfiguration. The initial guess shown in Figure 4.14 is generated by solving the inverse dynamics problem to determine a set of generalized forces necessary to move along a specified collision-free discrete trajectory. The resulting generalized forces have a similar impulsive quality to those in the puzzle example, as shown along side the DMOCC solution in Figure 4.13.



(a) Initial Guess, $\boldsymbol{\tau}^1$

(b) Initial Guess, $\boldsymbol{\tau}^J$

(c) DMOCC Solution, $\boldsymbol{\tau}^1$

(d) DMOCC Solution, $\boldsymbol{\tau}^J$

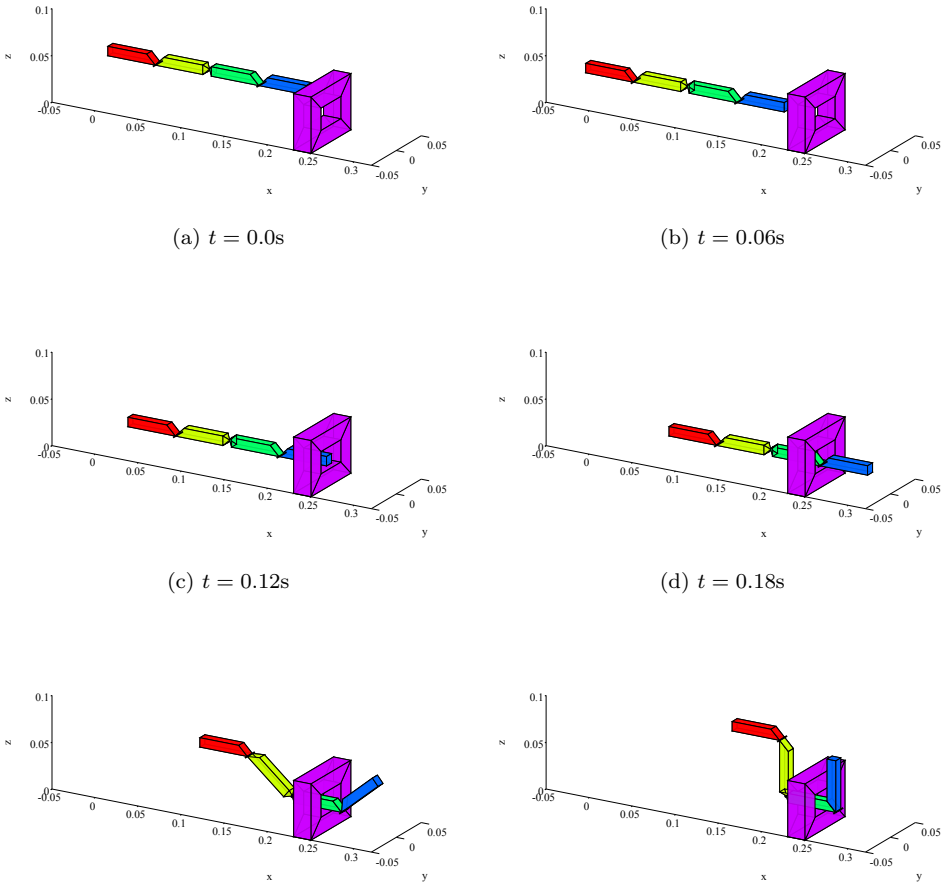
Figure 4.13: Comparison of generalized control forces in the inverse dynamics based initial guess and the final optimal solution for the articulated manipulator problem. Rapid variations in $\boldsymbol{\tau}^1$ are smoothed significantly in the DMOCC solution, and the magnitude of ths oscillations int $\boldsymbol{\tau}^J$ is also decreased.

As in the previous example, the initial and final momentum conditions are not met in the initial guess, however all other constraints are exactly satisfied. As previously mentioned, the final configuration is constrained by box constraints on the absolute position and orientation of body 3 and by the relative orientation of the bodies in the chain as follows

$$\begin{bmatrix} \boldsymbol{d}_1^{1\,T}\boldsymbol{d}_3^2 + 1 \\ \boldsymbol{d}_1^{2\,T}\boldsymbol{d}_3^3 - 1 \\ \boldsymbol{d}_3^{2\,T}\boldsymbol{d}_1^3 - 1 \\ \boldsymbol{d}_1^{3\,T}\boldsymbol{d}_3^4 + 1 \end{bmatrix} = \mathbf{0} \in \mathbb{R}^4.$$

Note that due to the kinematic linkages, only one additional constraint is needed to specify the final relative configuration of bodies linked by revolute joints, however two constratins are needed to specify the final relative condiguration of the bodies linked by a spherical joint. The time step used in this example is $h = 0.01$, and the problem was solved over thirty intervals, so that the total time of the maneuver is $t_K = 0.3s$ and $\boldsymbol{q}_d$ has $K + 1 = 31$ nodes. As shown in Figure 4.15, in addition to generating a relatively smooth maneuver as evidenced by the evolution of the tip location shown in Figure 4.16, the final configuration in the minimal control effort DMOCC problem is slightly askew and offset from the center of the hole when compared to the initial guess. However this still yields

a configuration from which the manipulator could proceed to pick up the block. Figure 4.17 shows that the DMOCC solution, which reduced the control effort by a factor of $\sim$100 from $9.412N \cdot s$ to $1.335e-2$N·s is indeed momentum-consistent. The reduction in control effort is due to the same reasons as the previous example.

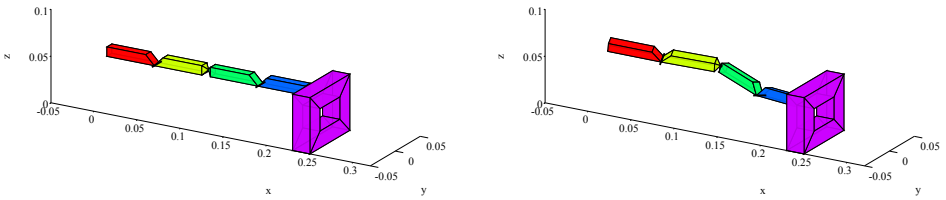


(a) $t = 0.0$s (b) $t = 0.06$s

(c) $t = 0.12$s (d) $t = 0.18$s

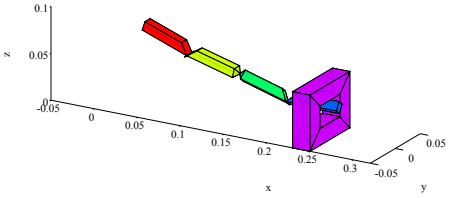
(e) $t = 0.24$s (f) $t = 0.30$s

Figure 4.14: Initial guess for the control of an articulated manipulator to prepare to pick up a block. Note that the final configuration condition is met, however the final momentum condition is not. The maneuver is realized by linking a series of constant velocity sub-trajectories.

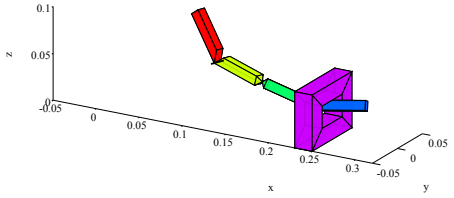


(a) $t = 0.0$s

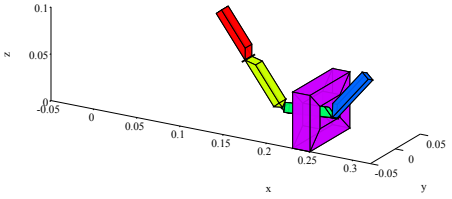
(b) $t = 0.06$s



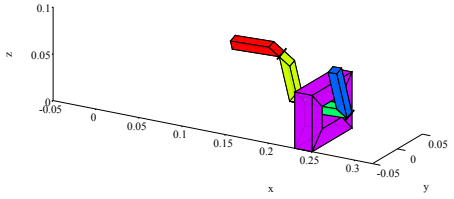
(c) $t = 0.12$s



(d) $t = 0.18$s



(e) $t = 0.24$s



(f) $t = 0.30$s

Figure 4.15: Results of the DMOCC problem for the control of an articulated manipulator to prepare to pick up a block. The minimial control effort manueuver subject to box constraints on the final aboslute configuration results in a final configuration in which the hook is slightly askew and offset from the center in the direction closest to the initial configuration.

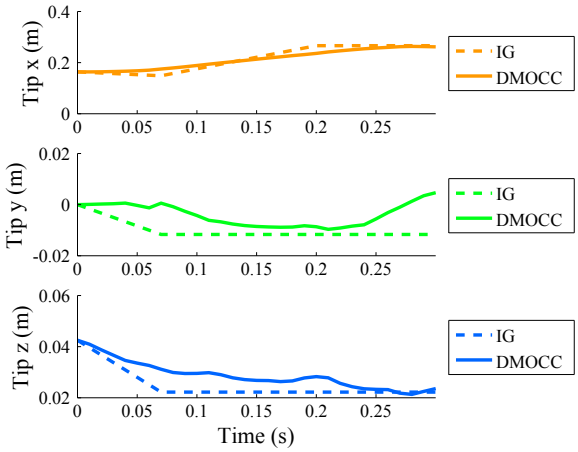


Figure 4.16: Comparison of the tip location as a function of time between the initial guess and DMOCC solution. The oscillations in the DMOCC solution are due to the manipulator avoiding collisions with the block at various points in the trajectory as it maneuvers in close proximity to the block.

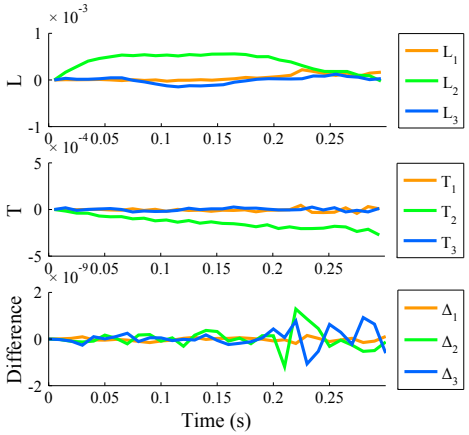


Figure 4.17: Momentum consistency of the optimal control solution for the control of the articulated manipulator. The tolerance on constraint violation used for the SQP solver in this problem is $1e$–9N·s and violations of the consistency condition are on the same order.

4.4 Optimal Control with Planned Contacts

In fully actuated systems with no bounds on the controls which are not overly-constrained from a kinematic perspective, the DMOCC problem can be used to find at least a local minimum which avoids all collisions given a reasonable initial guess. However, in more realistic situations involving bounded actuation and under-actuated problems (that is, $\dim \boldsymbol{\tau}_k < \dim \boldsymbol{u}_k$), it is likely that contact interactions will be unavoidable. In related situations, it is possible that the objective of the the controller is unreachable without making use of contact interactions in the solution. Furthermore, the goal of some controllers might be to achieve a certain contact configuration or a specific contact interaction, in which case it does not make sense to formulate the DMOCC problem with collision avoidance at every node.

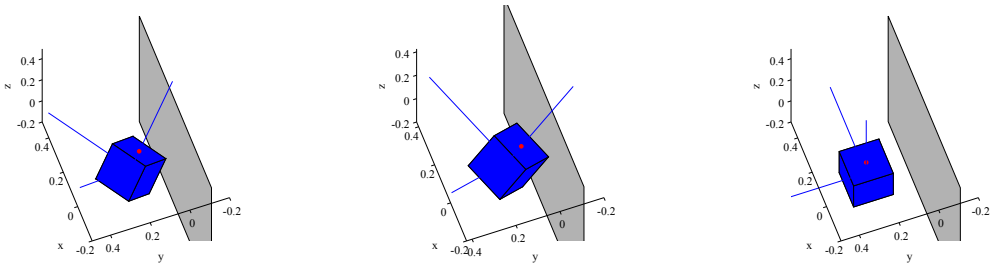
One option is to introduce contact into the dynamics via a penalty based approach, as described in Section 3.3. However, the disadvantages of the use of this method in forward dynamics problems (i.e., limits on time-steps, non-physical overlap) are compounded in the optimal control problem, as will be discussed in Section 4.4.1. A superior formulation–to be described in Section 4.4.3–is related to the DCR-DMC formulation for forward dynamics and makes use of the DMOCC framework to explicitly introduce collisions into the dynamics at a pre-seeded set of times nodes, $t_\iota, \iota = 1 \dots N_\iota$, by replacing the appropriate relationships between discrete momenta in the constraint set of (4.21) with relationships that reflect changes in momenta due to contact, which are closely related to those developed in Section 3.5. The cornerstone of the method is the introduction of an additional set of variables, $\sigma_\gamma, \gamma = 1 \dots N_\iota + 1$, which allow the time step on either side of the contact nodes to vary, thus preventing the problem from being over-constrained by prescribing exactly when contact takes place.

4.4.1 Penalty-Based Formulation

As in the forward dynamics problem, introducing elastic collisions into the DMOCC problem via a unilateral contact potential is the simplest approach from an implementation standpoint, as (4.21) can be used directly, along with (3.38), without any further modifications. However, the loss in accuracy due to overlap can be seen as more detrimental for the optimal control problem than the forward dynamics problem because the meaning of actuation forces in overlap configurations is suspect. Still, due to the simplicity of the approach, it is worth exploring at least a simple example to be compared to the more accurate approach in Section 4.4.3.

4.4.2 Example: Penalty-Based Formulation

Cube Hitting a Wall. The dynamics of a fully actuated cube interacting with a wall in a perfectly elastic collision modeled by the unilateral contact potential are explored. In this example, the contact parameter is $C = 20$, the total number of nodes 51 so that $K = 50$, and the base time step $h = 0.01s$, for a total simulation time of $t_K = 0.75$s. The initial position and velocity of the cube are fully specified with zero initial angular velocity, and the initial translational velocity towards the wall at a velocity of $\dot{\boldsymbol{\varphi}}_{\varphi}^{0} = -0.25\boldsymbol{e}_y$ m/s as shown in Figure 4.18, which also includes an intermediate configuration in which the cube penetrates into the wall, which can be seen more clearly in Figure 4.19. The goal of the maneuver is reach a final–again, fully specified–state in which the axes of the cube are aligned with the axes of the global reference frame, the angular velocity is zero, and the translational velocity is *away* and given by $\dot{\boldsymbol{\varphi}} = 0.25\boldsymbol{e}_y$ m/s. In addition, limits on the magnitude of the generalized discrete forces $\boldsymbol{\tau}_k$ are introduced so that it is not actually possible for the cube to reverse its momentum before it hits the wall. Specifically the magnitude of components of $\boldsymbol{\tau}_{\varphi}$ cannot exceed 0.002 and the magnitude of components of $\boldsymbol{\tau}_T$ cannot exceed 0.001. The objective is to minimize the total control effort of the maneuver, defined as the sum of the generalized forces squared.



(a) Initial Configuration

(b) Intermediate Configuration, $t = 0.27$s

(c) Final Configuration

Figure 4.18: Initial (Figure 4.18a) and final (Figure 4.18c) specified configurations for a rigid cube striking a wall, with directors shown. The initial specified velocities are $\boldsymbol{\omega}^0 = \mathbf{0}$ and $\dot{\boldsymbol{\varphi}}^0 = -0.25\boldsymbol{e}_y$, and the final specified velocities are $\boldsymbol{\omega}^K = \mathbf{0}$ and $\dot{\boldsymbol{\varphi}}^K = 0.25\boldsymbol{e}_y$. Figure 4.18b shows an (unspecified) intermediate configuration in which the cube and wall are overlapping.

The optimal solution for the generalized control forces, $\boldsymbol{\tau} \in \mathbb{R}^6$ is shown in Figure 4.20, and the momentum consistency of the solution is shown in Figure 4.21. Note that if changes in momentum due to the asymmetry of the contact potential (induced by the stationary wall with normal direction $\boldsymbol{e}_2$) are not accounted for, only the L_2 component of the angular momentum should be conserved. This problem was solved using Matlab's built in nonlinear constrained optimization routine, fmincon, with the SQP solver selected.

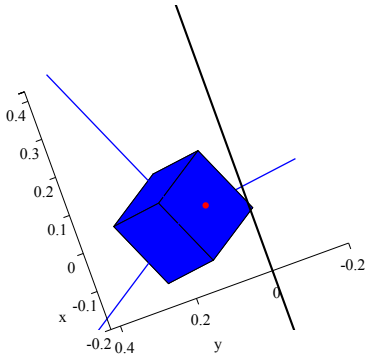
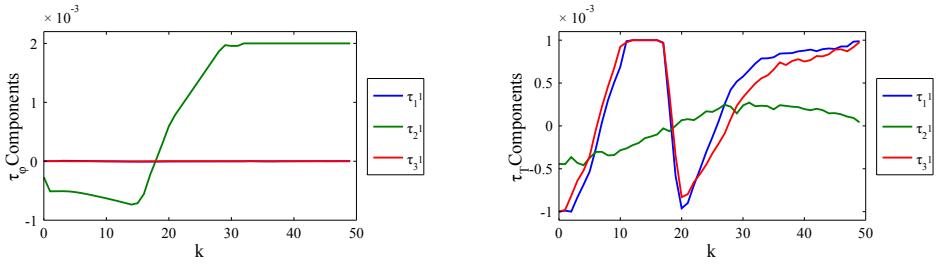


Figure 4.19: Close up view of the cube overlapping the wall at $t = 0.27s$.



(a) DMOCC Solution, $\boldsymbol{\tau}_{\varphi}$

(b) DMOCC Solution, $\boldsymbol{\tau}_{T}$

Figure 4.20: DMOCC solution for the generalized control forces to achieve the final desired configuration in the potential-based example of an actuated cube hitting a wall. Note that $\boldsymbol{\tau}_{\varphi_2}$ has the effect of first accelerating the cube towards the wall, and then resuming the acceleration away from the wall to achieve the final desired translational velocity away from the wall, while taking advantage of the presence of the wall to reverse the cube's translational velocity.

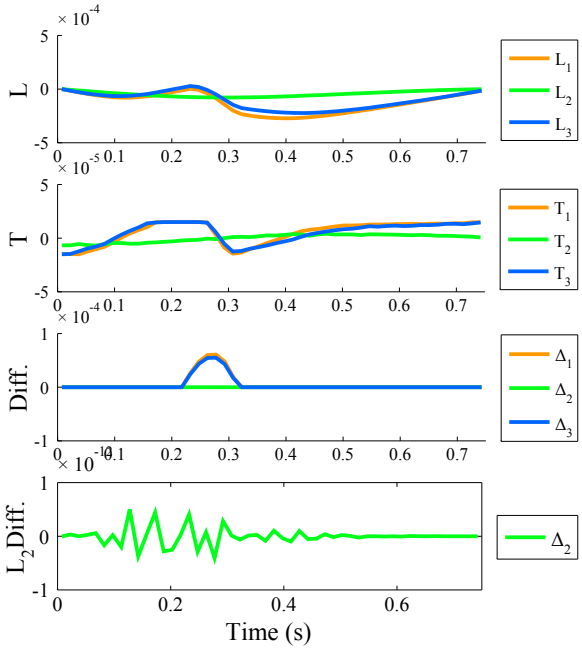


Figure 4.21: Angular momentum evolution, $\boldsymbol{L}$, net control torque, $\boldsymbol{T}$, and $\boldsymbol{\Delta}$, as defined in (4.12). The changes in angular momentum are consistent with the applied control torques, and the interaction with the stationary wall.

4.4.3 Momentum Reflection Formulation

In the momentum reflection formulation, the approximations made in the development of the integrator in Section 3.5 are explicitly enforced in the formulation of the DMOCC problem. That is, contacts are no longer *approximated* to take place at time nodes, the are *required* to take place at time nodes, $t_\iota, \iota = 1, \ldots, N_\iota$ via the introduction of a set of path constraints in the optimal control problem. The new description of the constrained discrete mechanics problem can be interpreted as 'daisy chaining' a series of $N_\iota + 1$ constant time step discrete trajectories together, and requiring that variations of the initial configuration of the $\iota + 1^{st}$ trajectory be equal to variations in the final configuration of the ι^{th} trajectory, i.e. $\delta(\boldsymbol{q}_N)_\iota = \delta(\boldsymbol{q}_0)_{\iota+1}$, rather than the standard statement that these variations vanish at the endpoints. As such, the initial momentum of the $\iota + 1^{st}$ trajectory is the post-contact final momentum of the ι^{th} trajectory.

In order to avoid over-constraining the system, this requirement is enabled by the introduction of $N_\iota + 1$ additional variables, σ_γ, which allow the time step to vary in each sub-trajectory on either side of the pre-seeded contact node set, thus allowing the *physical* contact time(s) to vary, as well as the total time of the maneuver, depending on how the constraint is stated.

For ease of notation, the following amended definitions of the discrete (augmented) Lagrangian and are introduced:

$$\overline{L}_{d,\sigma} = (\sigma h)\left[L\left(\frac{\boldsymbol{q}_{k+1}+\boldsymbol{q}_k}{2}, \frac{\boldsymbol{q}_{k+1}-\boldsymbol{q}_k}{\sigma h}\right)\right] - \frac{\sigma h}{2}\boldsymbol{G}^T(\boldsymbol{q}_{k+1})\boldsymbol{\lambda}_{k+1} - \frac{\sigma h}{2}\boldsymbol{G}^T(\boldsymbol{q}_k)\boldsymbol{\lambda}_k \tag{4.25}$$

$$\boldsymbol{f}^-_{k,\sigma} = \frac{\sigma h}{2}\boldsymbol{B}^T(\boldsymbol{q}_k)\boldsymbol{\tau}_k, \tag{4.26a}$$

$$\boldsymbol{f}^+_{k-1,\sigma} = \frac{\sigma h}{2}\boldsymbol{B}^T(\boldsymbol{q}_k)\boldsymbol{\tau}_k. \tag{4.26b}$$

In the preceding equations, the time step for a given trajectory is given by σh, with h as the so-called 'base' time step. Each sub-trajectory satisfies a separate discrete stationarity principle for its own action, $I_{d,\sigma}$, defined as the analog to (4.6), but with non-vanishing variations at the endpoints of the sub-trajectories with the exception of the first configuration of the first sub-trajectory, and the last configuration of the last sub-trajectory.

The resulting expressions to be rendered stationary at the endpoints of the intervals are

$$\left(D_2 L_{d,\sigma_i}(\boldsymbol{q}_{\iota-1}, \boldsymbol{q}_\iota^+) - \frac{\sigma_i h}{2}\boldsymbol{G}^T(\boldsymbol{q}_\iota^+)\boldsymbol{\lambda}_\iota^+ + \boldsymbol{f}_{\iota-1,\sigma_i}^+\right)\delta\boldsymbol{q}_\iota^+.$$

in the pre-contact interval and

$$\left(D_1 L_{d,\sigma_{i+1}}(\boldsymbol{q}_\iota^-, \boldsymbol{q}_{\iota+1}) - \frac{\sigma_{i+1} h}{2}\boldsymbol{G}^T(\boldsymbol{q}_\iota^-)\boldsymbol{\lambda}_\iota^- + \boldsymbol{f}_{\iota,\sigma_{i+1}}^-\right)\delta\boldsymbol{q}_\iota^-.$$

in the post-contact interval, with the additional restriction that $\delta\boldsymbol{q}_\iota^\pm \in T\partial\mathcal{A}$. To link the sub-trajectories and thus render the total forced discrete action sum, $\sum_{i=1}^{N_\iota+1} I_{d,\sigma_i}$ stationary, it must be true that $\boldsymbol{q}_\iota^+ = \boldsymbol{q}_\iota^-$, and $\delta\boldsymbol{q}_\iota^+ = \delta\boldsymbol{q}_\iota^-$, and likewise for the Lagrange multipliers, i. e. , $\boldsymbol{\lambda}_\iota^+ = \boldsymbol{\lambda}_\iota^-$, and $\delta\boldsymbol{\lambda}_\iota^+ = \delta\boldsymbol{\lambda}_\iota^-$, so that stationarity with respect to the contact configuration is given by

$$D_2 L_{d,\sigma_i}(\boldsymbol{q}_{\iota-1}, \boldsymbol{q}_\iota) + D_1 L_{d,\sigma_{i+1}}(\boldsymbol{q}_\iota, \boldsymbol{q}_{\iota+1}) + \boldsymbol{f}_{\iota-1,\sigma_1}^+ + \boldsymbol{f}_{\iota,\sigma_{i+1}}^- - \frac{h(\sigma_i + \sigma_{i+1})}{2}\boldsymbol{G}(\boldsymbol{q}_\iota)^T\lambda_\iota = \boldsymbol{0}$$

along with $\delta\boldsymbol{q}_\iota \in T\partial\mathcal{A}$. In the analog to Equation 3.68, this can be re-expressed as

$$D_2 L_{d,\sigma_i}(\boldsymbol{q}_{\iota-1}, \boldsymbol{q}_\iota) + D_1 L_{d,\sigma_{i+1}}(\boldsymbol{q}_\iota, \boldsymbol{q}_{\iota+1}) + \boldsymbol{f}_{\iota-1,\sigma_i}^+ + \boldsymbol{f}_{\iota,\sigma_{i+1}}^- + \frac{h(\sigma_i + \sigma_{i+1})}{2}\left[\boldsymbol{G}(\boldsymbol{q}_\iota)^T\ \nabla g(\boldsymbol{q}_\iota)\right]\boldsymbol{\mu}_{c,\iota} = \boldsymbol{0}.$$

Making use of the forced discrete Legendre transforms

$$\begin{aligned}
&\mathbb{F}_\sigma^{cf+} : (\boldsymbol{q}_{\iota-1}, \boldsymbol{q}_\iota, \sigma) \mapsto (\boldsymbol{q}_\iota, \boldsymbol{p}_\iota^+)\\
&\boldsymbol{p}_\iota^+ = D_2 L_{d,\sigma_i}(\boldsymbol{q}_{\iota-1}, \boldsymbol{q}_\iota^+) - \frac{\sigma_i h}{2}\boldsymbol{G}^T(\boldsymbol{q}_\iota^+)\boldsymbol{\lambda}_\iota^+ + \boldsymbol{f}_{\iota-1,\sigma_i}^+\\
&\mathbb{F}_\sigma^{cf-} : (\boldsymbol{q}_\iota, \boldsymbol{q}_{\iota+1}, \sigma) \mapsto (\boldsymbol{q}_\iota, \boldsymbol{p}_\iota^-)\\
&\boldsymbol{p}_\iota^- = -D_1 L_{d,\sigma_{i+1}}(\boldsymbol{q}_\iota^-, \boldsymbol{q}_{\iota+1}) + \frac{\sigma_{i+1} h}{2}\boldsymbol{G}^T(\boldsymbol{q}_\iota^-)\boldsymbol{\lambda}_\iota^- - \boldsymbol{f}_{\iota,\sigma_{i+1}}^-
\end{aligned}$$

results in

$$\boldsymbol{p}_\iota^+ - \boldsymbol{p}_\iota^- = \frac{h(\sigma_i + \sigma_{i+1})}{2}\mu_{c,\iota}\nabla g(\boldsymbol{q}_\iota),$$

or, in the reduced form,

$$ {}^{P}\boldsymbol{p}_{\iota}^{+} - {}^{P}\boldsymbol{p}_{\iota}^{-} = \frac{h(\sigma_i + \sigma_{i+1})}{2} \mu_{c,\iota} \, {}^{P}\nabla g(\boldsymbol{q}_{\iota}), $$

with ${}^{P}\mathbb{F}_{\sigma}^{cf+}$ and ${}^{P}\mathbb{F}_{\sigma}^{cf-}$ defined by

$$ \begin{aligned} {}^{P}\boldsymbol{p}_{\iota}^{+} &= \boldsymbol{P}(\boldsymbol{q}_{\iota}^{+})^{T} \left[D_2 L_{d,\sigma_i}(\boldsymbol{q}_{\iota-1}, \boldsymbol{q}_{\iota}^{+}) + \boldsymbol{f}_{\iota-1,\sigma_i}^{+} \right] \\ {}^{P}\boldsymbol{p}_{\iota}^{-} &= \boldsymbol{P}(\boldsymbol{q}_{\iota}^{-})^{T} \left[-D_1 L_{d,\sigma_{i+1}}(\boldsymbol{q}_{\iota}^{-}, \boldsymbol{q}_{\iota+1}) - \boldsymbol{f}_{\iota,\sigma_{i+1}}^{-} \right]. \end{aligned} $$

Using the momentum decomposition strategies developed in Section 3.5, this relationship can be stated in a fully explicit form that does not depend on the Lagrange multiplier μ_c. In general, the initial projected momenta of the post-collision trajectory and the final projected momenta of the pre-collision trajectory are related by

$$ {}^{P}\boldsymbol{p}_{\iota}^{-} = \boldsymbol{F}_{P,mr}({}^{P}\boldsymbol{p}_{\iota}^{+}), $$

with

$$ \boldsymbol{F}_{P,mr}({}^{P}\boldsymbol{p}_{\iota}^{+}) = {}^{P}\boldsymbol{p}_{\iota}^{+} + \boldsymbol{I}_{norm}({}^{P}\boldsymbol{p}_{\iota}^{+}) + \boldsymbol{I}_{slide,P}({}^{P}\boldsymbol{p}_{\iota}^{+}), $$

according to the definitions in Section 3.5.4, whereby elastic, inelastic, and frictional collisions may all be considered in the DMOCC problem. This leads to a fully momentum-consistent solution to the optimal control problem, in which any change to the discrete momenta within each sub-trajectory is exactly due to the applied forces on the system and any asymmetric potentials present, and the relationship between subtrajectories precisely defined by momentum changes due to contact.

For contact node set I_{ι}, the DMOCC problem with collisions is then given by

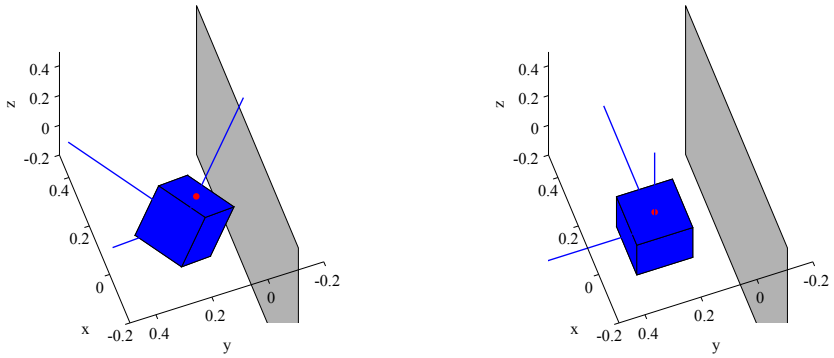
$$
\begin{aligned}
&\min_{\boldsymbol{u}_d,\boldsymbol{\tau}_d} \sum_{k=0}^{K-1} \bar{B}_d(\boldsymbol{u}_k, \boldsymbol{u}_{k+1}, \boldsymbol{\tau}_k, \sigma_i) \\
&\quad \text{subject to:} \\
&\quad {}^P\boldsymbol{p}_k^+ - {}^P\boldsymbol{p}_k^- = \mathbf{0}, k \notin I_\iota \\
&\quad {}^P\boldsymbol{p}_k^- = \boldsymbol{F}_{P,mr}({}^P\boldsymbol{p}_k^+), k \in I_\iota, \\
&\quad g(\boldsymbol{q}_k) = 0, k \in I_\iota, \\
&\quad g(\boldsymbol{q}_k) < 0, k \notin I_\iota, \\
&\quad \boldsymbol{s}_0(\boldsymbol{u}_0, \boldsymbol{u}_1, \boldsymbol{\tau}_0, \boldsymbol{q}^0, \dot{\boldsymbol{q}}^0) = \mathbf{0}, \\
&\quad \boldsymbol{s}_K(\boldsymbol{u}_{K-1}, \boldsymbol{u}_K, \tau_{K-1}, \boldsymbol{q}^K, \dot{\boldsymbol{q}}^K) = \mathbf{0}, \\
&\quad \boldsymbol{h}(\boldsymbol{q}_k, {}^P\boldsymbol{p}_k, \boldsymbol{\tau}_k) \leq \mathbf{0}, \\
&\quad \boldsymbol{w}(\sigma_\gamma, t_K) \leq \mathbf{0}, \\
&\quad \text{for } k = 1 \ldots K-1,\ \iota = 1 \ldots N_\iota + 1,\ N_\iota = |I_\iota|.
\end{aligned}
\tag{4.27}
$$

Note that the first set of constraints are valid away from the contact node set due to the introduction of the path constraints $g(\boldsymbol{q}_k) < 0, k \notin I_\iota$, which means that collisions are avoided at all other points in solution trajectory, and the $\sigma-$dependent version of the reduced forced discrete Legendre transform has been used. Furthermore, the function $\boldsymbol{w}(\sigma_\gamma, t_K)$ is used to–at minimum–prevent σ_γ from approaching 0, or to general constrain the total time of the maneuver e. g. by placing upper and lower bounds on σ_i, or by constraining the total time of the maneuver.

4.4.3.1 Examples: Momentum Reflection Formulation

The following examples illustrate the robustness and flexibility of the momentum reflection formulation of the DMOCC problem with contact (4.27) in designing maneuvers or trajectories for a variety of hybrid systems of non-smooth bodies that involve at least one contact interaction. Unless otherwise specified, all initial guesses are generated by an unactuated forward dynamics simulation with a constant time step and collisions treated by closest point projection and momentum decomposition, as described in Section 3.5. Solutions to the optimal control problem are found using Matlab's built in nonlinear constrained optimization sequential quadratic programming (SQP) active-set routine.

Cube Hitting a Wall. As a first example to explore the influence of the chosen contact node set, I_ι, on the final solution, the dynamics of a cube with one perfectly elastic collision with a wall are explored. In this example, $N_\iota = 1$, so $\{\sigma_\gamma\} = \{\sigma_1, \sigma_2\}$, and $\boldsymbol{\tau}_k \in \mathbb{R}^6$. The total number of nodes is 31 so that $K = 30$, and the base time step $h = 0.025s$. The initial position and velocity of the cube are fully specified with zero initial angular velocity, and an initial translational velocity is $\dot{\boldsymbol{\varphi}}^0 = -0.25\boldsymbol{e}_y$ m/s towards the wall, as shown in Figure 4.22. The goal of the maneuver is to reach a final–again, fully specified–state in which the axes of the cube are aligned with the axes of the global reference frame, the angular velocity is zero, and the translational velocity is *away* from the wall and given by $\dot{\varphi}^K = 0.25\boldsymbol{e}_y$ m/s. In addition, limits on the magnitude of the generalized discrete forces $\boldsymbol{\tau}_k$ are introduced, so that it is not actually possible for the cube to reverse its momentum before it hits the wall. Both σ_1 and σ_2 are initially set to 1, and the total time of the maneuver is fixed according to $w(\sigma_1, \sigma_2) = (\iota\sigma_1 + (K - \iota)\sigma_2)\, h - t_K$, with $t_K = 0.75s$–the same duration as the initial guess. The objective is to minimize the total control effort of the maneuver, defined as the sum of the generalized forces squared.



(a) Initial Configuration (b) Final Configuration

Figure 4.22: Initial and final specified configurations for a rigid cube striking a wall, with directors shown. The initial specified velocities are $\boldsymbol{\omega} = \mathbf{0}$ and $\dot{\boldsymbol{\varphi}} = -0.25\boldsymbol{e}_y$, and the final specified velocities are $\boldsymbol{\omega} = \mathbf{0}$ and $\dot{\boldsymbol{\varphi}} = 0.25\boldsymbol{e}_y$.

The DMOCC problem was solved for three different contact node sets, given by $I_\iota = \{4\}, \{9\}$, or $\{14\}$. The evolution of the total momentum, for the optimal solution with $I_\iota = \{9\}$ is shown in Figure 4.23.

This example is also used to check the sensitivity of the proposed method to the choice of contact set, I_ι, of course requiring that N_ι be the same for all comparisons. Figure 4.24 shows the qualitative equivalence between DMOCC solutions with three different choices of I_ι. In particular, the *physical* contact time remains essentially unchanged between the solutions, and the evolution of the control torques and the total control effort are similar (see Table 4.1).

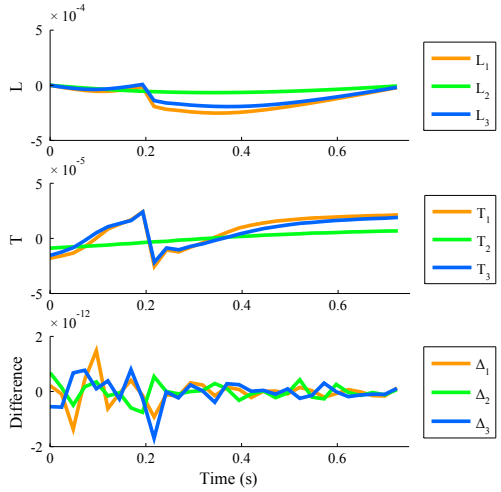


Figure 4.23: Angular momentum evolution, $\boldsymbol{L}$, net control torque $\boldsymbol{T}$, and $\boldsymbol{\Delta}$ as defined in (4.12). Note that the (expected) change in angular momentum due to the collision with the stationary wall has been added back in to illustrate the overall momentum consistency.

ι	4	9	14
σ_1	2.1795	0.9653	0.6186
σ_2	0.8185	1.0149	1.3337
t_ι	0.2180	0.2172	0.2156
J_d	$1.0244{\cdot}10^{-6}$	$1.0194{\cdot}10^{-6}$	$1.0005{\cdot}10^{-6}$

Table 4.1: Comparison between DMOCC solutions with $\iota = 4, 9, 14$ for the cube hitting wall example.

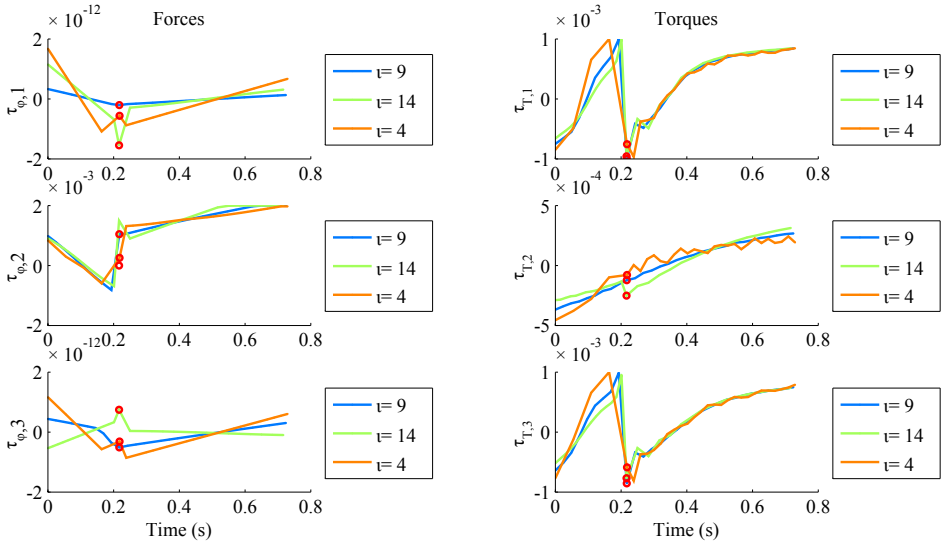
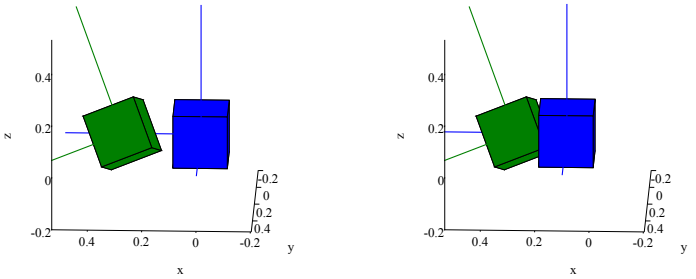


Figure 4.24: Evolution of generalized control forces plotted on preceding time node (i.e. $\boldsymbol{\tau}_k$'s value is plotted at t_k), with contact nodes highlighted in red.

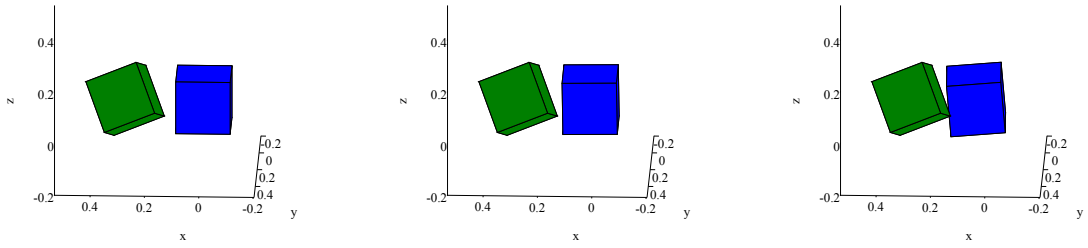
Two Colliding Cubes. In this example, as in the previous example, there is no way for collisions to be avoided in the solution to the optimal control problem. However now, rather than bounds on the control forces and torques being the limiting factor, there is a second unactuated body precluding the final desired configuration from being reached (see Figure 4.25), with the objective of minimizing the control effort. The total number of nodes is $K+1 = 51$, and $I_\iota = \{20, 18\}$. As in the previous example to total time of the maneuver is restricted by $w(\sigma_1, \sigma_2, \sigma_3) = (\iota_1\sigma_1 + (\iota_2 - \iota_1)\sigma_2 + (K - \iota_1 - \iota_2)\sigma_3) - t_K$, with $t_K = 0.5s$. The initial and final states of the blue cube (the only cube with actuation) are fully specified. In the final configuration, the blue cube is at rest, whereas in the initial configuration, the blue cube has a velocity of $\dot{\boldsymbol{\varphi}} = 0.25\boldsymbol{e}_x$ in the direction of the green cube, and zero angular velocity. Several snapshots of the motion are shown in Figure 4.26, and the evolution of angular momentum due to applied control forces is shown in Figure 4.27.



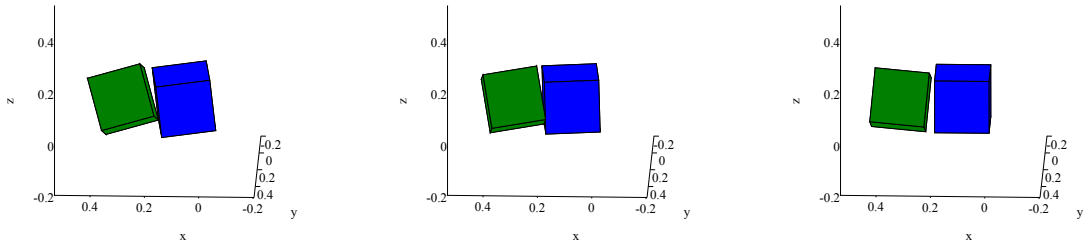
(a) Initial Configuration (b) Final (Inadmissible) Configuration

Figure 4.25: Initial and final specified configurations for the actuated blue cube, with directors shown. The initial specified velocities are $\boldsymbol{\omega} = \mathbf{0}$ and $\dot{\boldsymbol{\varphi}} = 0.25\hat{\boldsymbol{e}}_x$, and the final specified velocities are $\boldsymbol{\omega} = \mathbf{0}$ and $\dot{\boldsymbol{\varphi}} = \mathbf{0}$. Note that the final specified configuration is inadmissible if the unactuated green cube is not first pushed out of the way.

At the solution, $\sigma_1 = 0.9876$, $\sigma_2 = 1.013$, and $\sigma_3 = 1.000$, and the final control effort is $J_d = 2.857 \cdot 10^{-5}$.



(a) t = 0s (b) t = 0.089s (c) t = 0.188s



(d) t = 0.294s (e) t = 0.395s (f) t = 0.500s

Figure 4.26: Time lapse images of a planned motion involving two collisions between two cubes.

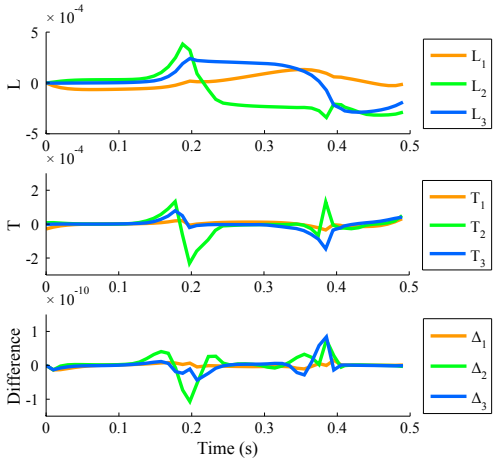


Figure 4.27: Angular momentum evolution, $\boldsymbol{L}$, net control torque, $\boldsymbol{T}$, and $\boldsymbol{\Delta}$, as defined in (4.12). The changes in angular momentum are consistent with the applied control torques.

Throwing Dice. This final example illustrates the present method's usefulness not only for systems involving contact interactions, but also for hybrid systems in which joints enter and leave the dynamics. The maneuver to be planned here is the idealized throwing of a die against a perfectly elastic wall before it lands on a sticky plastic floor, as shown in Figure 4.28. There is no actuation in the system, but forcing due to gravity is present. The goal of the optimization is to determine an initial state of the system (within bounds) that allow for a final configuration with a specified face pointing up and momentum minimized to be reached; i.e., the objective function is given by

$$J_d = (\boldsymbol{d}_3^T \boldsymbol{e}_z - 1)^2 + ({}^P\boldsymbol{p}_K^+)^T({}^P\boldsymbol{p}_K^+).$$

The contact set is given by $I_\iota = \{20, 22, 40\}$, with the first two nodes corresponding to elastic collisions with the wall, and the final a perfectly plastic collision with the floor, after which the die sticks to the floor. The two elastic collisions with the wall are treated in the normal way, and the sticky floor modeled by the combination of a sticky perfectly plastic collision and the addition of a spherical joint at the first corner to hit the floor. That is, at $\iota = 40$, the relationship to be enforced is given by:

$${}^{P^S}\boldsymbol{p}_\iota^- = {}^{P^S}(\boldsymbol{p}_\iota^{S,+}),$$

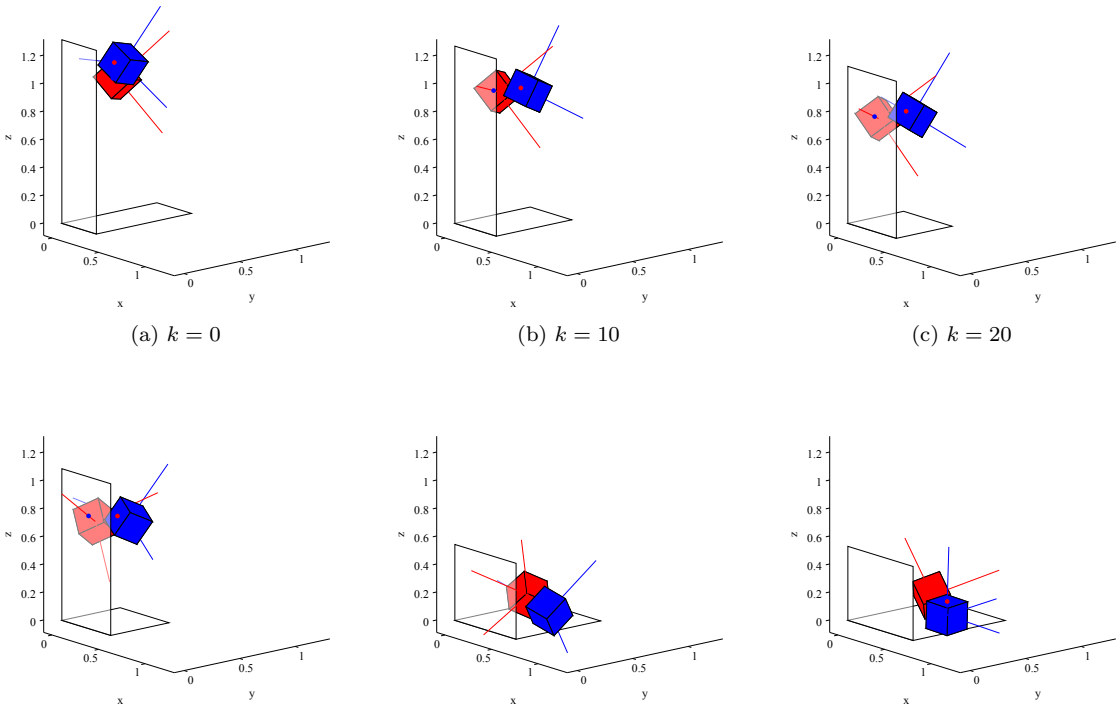
in which

$$\boldsymbol{p}_\iota^{S,+} = \boldsymbol{M}\boldsymbol{P}_{int}\boldsymbol{M}_{red}^{-1}\left({}^P\boldsymbol{p}_\iota^+ - {}^P\boldsymbol{p}_{\iota,norm}^+ - {}^P\boldsymbol{p}_{\iota,slide}^+\right)$$

represents plastic, sticky, frictional collision, with ${}^P\boldsymbol{p}_{\iota,norm}$ and ${}^P\boldsymbol{p}_{\iota,norm}$ defined in Section 3.5, and $\boldsymbol{P}^S$ the null space matrix due to the spherical joint. The post-collision generalized momenta $\boldsymbol{p}_\iota^{S,+}$, irrespective of the addition of a joint to the system, lead to no relative motion between the contact point and the floor, and are thus consistent with admissible joint velocities due to the addition of the spherical joint at the contact point. The location of the joint is determined in the course of the optimization by the location of first corner to hit the floor, as determined by the SSH LP.

Figure 4.28 shows a comparison between the initial guess and DMOCC solution at various points in the trajectory. In the first part of the motion–i.e., before the die strikes the floor–the L_3 component

of the total angular momentum is conserved* away from collisions with the wall, as illustrated in Figure 4.29. The addition of the spherical joint to the stationary floor breaks the remaining symmetry in the system. At the solution $\sigma_1 = 0.8634$, $\sigma_2 = 1.3788$, $\sigma_3 = 1.1093$, and $\sigma_4 = 1.029$. A base time step of $h = 0.01$, is used.



(a) $k = 0$ (b) $k = 10$ (c) $k = 20$

(d) $k = 22$ (e) $k = 40$ (f) $k = 50$

Figure 4.28: Time lapse images of optimized dice throwing comparing the initial guess (red with blue dot) with the final solution (blue with red dot) at key time nodes.

*That is, this is the expected result if only control forces and not $\boldsymbol{T}_{grav}$ are accounted for in $\boldsymbol{\Delta}$.

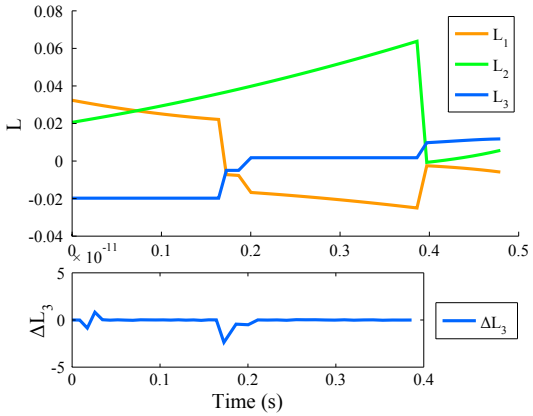


Figure 4.29: Angular momentum evolution, $\boldsymbol{L}$ through the optimal motion of the die. Due to the presence of gravity and collisions with the stationary wall and floor, $\boldsymbol{L}$ is not fully conserved throughout the trajectory. Removing changes in L_3 due to the collision with the wall shows that it is conserved prior to the addition of the spherical joint to the system.

4.5 Feedback Stabilized Trajectory Tracking

This section differs from the preceding sections in this chapter in that the end goal is *not* to develop a method for determining the optimal control of the spacecraft, but rather to develop a feedback stabilized controller which can be used to reliably attain a desired state. The controller described in this section, which uses a minimal parametrization of the state error, could be used, for example, to track or correct to a pre-computed optimal trajectory, e.g. in the presence of disturbances. In this section, a general feedback stabilization control law is considered which will shown to be globally asymptotically stable (in the sense of Lyapunov) to a desired trajectory for fully actuated spacecrafts. The strategy for choosing such a trajectory will also be discussed. To this end, the form of the fully actuated control law is motivated by the under-actuated nature of the electromagnetic actuation system being designed for the AAReST mission, and by certain characteristics of magnetic actuation–for example the fact that magnetic moments align with magnetic fields, and magnetic force directions are determined by the orientation of the moment. Ongoing work is focused on characterizing the unique magnetic actuation system (see Appendix B) on the AAReST CubeSats and extending the methodology to a docking maneuver executed with that actuation system.

4.5.1 Spacecraft Dynamics

Each spacecraft (SC) configuration is described by its orientation matrix $\boldsymbol{R} \in SO(3)$ and position vector $\boldsymbol{\varphi} \in \mathbb{R}^3$ with respect to frame $\mathcal{F}$. The angular velocity is denoted $\boldsymbol{\omega} \in \mathbb{R}^3$ in the body frame, and the linear velocity is $\boldsymbol{v} \in \mathbb{R}^3$ in frame $\mathcal{F}$. The full state us collectively denoted by $\boldsymbol{s} = (R, \boldsymbol{q}, \boldsymbol{\omega}, \boldsymbol{v}) \in S$, where $S = SO(3) \times \mathbb{R}^3 \times \mathbb{R}^3 \times \mathbb{R}^3$ is the state space. The mass alternately be noted as a scalar M or the diagonal matrix $\boldsymbol{M} = M\boldsymbol{I}_{3\times 3}$ and the inertia matrix by $\boldsymbol{J} \in \mathbb{R}^{3\times 3}$. The equations of motion are

$$\dot{\boldsymbol{R}} = \boldsymbol{R}\widehat{\boldsymbol{\omega}}, \tag{4.28a}$$

$$\dot{\boldsymbol{\varphi}} = \boldsymbol{v}, \tag{4.28b}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = \boldsymbol{J}\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau}_c + \boldsymbol{\tau}_{\text{ext}}(\boldsymbol{s}), \tag{4.28c}$$

$$\boldsymbol{M}\dot{\boldsymbol{v}} = \boldsymbol{f}_c + \boldsymbol{f}_{\text{ext}}(\boldsymbol{s}), \tag{4.28d}$$

where as before, $\widehat{\cdot}: \mathbb{R}^3 \to \mathfrak{so}(3)$ is the skew-symmetric map

$$\widehat{\boldsymbol{\omega}} = \begin{bmatrix} 0 & -w^3 & w^2 \\ w^3 & 0 & -w^1 \\ -w^2 & w^1 & 0 \end{bmatrix},$$

and $\boldsymbol{\tau}_c$ and $\boldsymbol{f}_c$ represent control torques and forces. The external torques $\boldsymbol{\tau}_{\text{ext}}$ and forces $\boldsymbol{f}_{\text{ext}}$ represent standard orbital perturbations, e.g. due to oblateness, atmospheric drag or solar pressure (Wieland [85], Wiesel [86] are two of many useful references on the topic).

4.5.2 Trajectory Tracking

The goal is to track a position curve in $\mathbb{R}^3$ that will smoothly guide the spacecraft towards docking. In this section, the notation $(\cdot)_d$ denotes components of the *desired* state. This position curve, $\boldsymbol{\varphi}_d : [0, \infty] \to \mathbb{R}^3$, also determines a direction of motions $\dot{\boldsymbol{\varphi}}_d(t)$ at any time t. To achieve alignment, $\boldsymbol{\varphi}_d$ is tracked and at the same time the docking of the spacecraft are oriented along the unit direction $\boldsymbol{d}_{1,d}(\boldsymbol{\varphi}_d(t))$ and rotate the spacecraft around $\boldsymbol{d}_{1,d}(\boldsymbol{\varphi}_d(t))$ to match the receiving docking part.

More formally, assume that docking happens at configuration $(\tilde{\boldsymbol{R}}_0, \tilde{\boldsymbol{\varphi}}_0)$. Thus $\boldsymbol{\varphi}_d$ is chosen to satisfy

$$\lim_{t \to T} \boldsymbol{\varphi}_d(t) = \tilde{\boldsymbol{\varphi}}_0, \qquad \lim_{t \to T} \boldsymbol{d}_{1,d}(t) = \bar{\dot{\boldsymbol{q}}}_d(t), \qquad \tilde{\boldsymbol{d}}_{1,d} = \bar{\dot{\boldsymbol{\varphi}}}_d(T), \tag{4.29}$$

for some finite T, where $\boldsymbol{d}_i$ for $i = 1, 2, 3$ represent the the i-th standard basis vector of $\mathbb{R}^3$ rotated by the body's orientation $\boldsymbol{R}$. These vectors correspond to the columns of $\boldsymbol{R}$ and are the same as the directors in the redundant formulation. Here it is assumed that the spacecraft has non-zero velocity $\dot{\boldsymbol{q}}_d(T)$ immediately before docking. This velocity is necessary to, e. g., engage a mechanical latch.

In addition, a proper angular alignment, i.e. $\boldsymbol{R}(T) = \tilde{\boldsymbol{R}}_0$ is achieved by aligning one of the remaining axes ($i = 2$ or 3) to the direction $\boldsymbol{d}_{i,d} = \bar{\boldsymbol{a}}_i$ defined by

$$\begin{aligned} \boldsymbol{a}_i &= \tilde{\boldsymbol{R}}_0 \boldsymbol{e}_i - (\boldsymbol{d}_{1,d} \cdot \tilde{\boldsymbol{R}}_0 \boldsymbol{e}_i) \boldsymbol{d}_{1,d}, \\ \bar{\boldsymbol{a}}_i &= \frac{\boldsymbol{a}_i}{\|\boldsymbol{a}_i\|} \end{aligned} \tag{4.30}$$

so that $\bar{\boldsymbol{a}}$ is the projection of $\tilde{\boldsymbol{R}}_0 \boldsymbol{e}_i$ onto the current desired frame. The full orientation along the trajectory $\boldsymbol{R}_d(t)$ is now completely determined.

4.5.3 State Error

We start by defining the position error in frame $\mathcal{F}$

$$\boldsymbol{e}_{\varphi} = \boldsymbol{\varphi} - \boldsymbol{\varphi}_d \qquad : \text{range error} \tag{4.31}$$

$$\boldsymbol{e}_{v} = \boldsymbol{v} - \dot{\boldsymbol{\varphi}}_d \qquad : \text{velocity error} \tag{4.32}$$

The next step is to define the desired orientation $\boldsymbol{R}_d$. For concreteness, we set $i = 2$ in (4.30), and let

$$\boldsymbol{R}_d = [\boldsymbol{d}_{1,d} \mid \bar{\boldsymbol{a}}_2 \mid \boldsymbol{d}_{1,d} \times \bar{\boldsymbol{a}}_2] \,.$$

It is convenient to describe the error in the orientation and angular velocity of the spacecraft in the spacecraft's own body frame for several reasons. First, for autonomous crafts, this is typically the frame in which the spacecraft measures its own state, and second, the actuator properties are readily characterized in the body frame, so that desired forces and torques can be calculated and generated intuitively. The error in orientation expressed as $\boldsymbol{R}_e \in SO(3)$–in the current body frame–is then expressed as

$$\begin{aligned} \boldsymbol{R}_e &= \boldsymbol{R}^T \left[\boldsymbol{d}_{1,d} \mid \bar{\boldsymbol{a}}_2 \mid \boldsymbol{d}_{1,d} \times \bar{\boldsymbol{a}}_2\right] \\ &= \boldsymbol{R}^T \boldsymbol{R}_d. \end{aligned}$$

In a given configuration, the spacecraft should be turning to either attain or stay on the desired trajectory. The desired angular velocity–expressed in the reference frame of the desired orientation–is then

$$\widehat{\boldsymbol{\omega}}_d = \boldsymbol{R}_d^T \dot{\boldsymbol{R}}_d$$

We can now define the orientation error terms

$$\boldsymbol{e}_R = \vartheta^{-1}(\boldsymbol{R}_e), \qquad : \text{orientation error} \tag{4.33}$$

$$\boldsymbol{e}_{\omega} = \boldsymbol{\omega} - \mathrm{Ad}_{R_e} \boldsymbol{\omega}_d \qquad : \text{angular velocity error.} \tag{4.34}$$

In (4.33), $\vartheta : \mathfrak{g} \to G$ is a *retraction map* from a Lie algebra $\mathfrak{g}$ to its Lie group G. In this case, $G = SO(3)$ and $\mathfrak{g} = so(3)$. In addition, the operator $\mathrm{Ad}_g : \mathfrak{g} \to \mathfrak{g}$ defined as

$$\operatorname{Ad}_g \xi = g\xi g^{-1}$$

for $\xi \in \mathfrak{g}$ and can be regarded as a change of basis with respect to argument $g \in G$.

Definition 4.1. *Given a map $\vartheta : \mathfrak{g} \to G$, its* right-trivialized tangent $d\vartheta_\xi : \mathfrak{g} \to \mathfrak{g}$ *and its* inverse $d\vartheta_\xi^{-1} : \mathfrak{g} \to \mathfrak{g}$ *are such that, for some $g = \vartheta(\xi) \in G$ and $\mu \in \mathfrak{g}$, the following holds (c.f. [57])*

$$\partial_\xi \vartheta(\xi) \cdot \mu = d\vartheta_\xi(\mu) \cdot \vartheta(\xi) \tag{4.35}$$

$$\partial_\xi \vartheta^{-1}(g) \cdot \mu = d\vartheta_\xi^{-1}\left(\mu \cdot \vartheta(-\xi)\right). \tag{4.36}$$

In the preceding equations, $\partial_\xi \vartheta(\xi) \cdot \mu$ denotes the derivative taken in the direction μ, and $\zeta \cdot \vartheta(\cdot)$ denotes the right action of $\vartheta(\cdot)$ on $\zeta \in \mathfrak{g}$.

Several options are available to define the retraction map, ϑ, however in this work consideration will be restricted to the exponential map and the Cayley map. The exponential map, $\exp : \mathfrak{g} \to G$ defined by $\exp \xi = \gamma(1)$, where $\gamma : \mathbb{R} \to G$ is the integral curve through the identity of the vector field associated with $\xi \in \mathfrak{g}$, with $\dot{\gamma}(0) = \xi$.

Definition 4.2. *The right-trivialized derivative of the* exp *map and its inverse are defined as*

$$d\exp_x y = \sum_{j=0}^{\infty} \frac{1}{(j+1)!} \operatorname{ad}_x^j y \tag{4.37a}$$

$$d\exp_x^{-1} y = \sum_{j=0}^{\infty} \frac{B_j}{j!} \operatorname{ad}_x^j y \tag{4.37b}$$

where $\operatorname{ad}_x y$ is the Lie bracket operator $\operatorname{ad} : \mathfrak{g} \times \mathfrak{g} \to \mathfrak{g}$, and is defined as

$$\operatorname{ad}_x y = xy - yx,$$

with $\operatorname{ad}_x^0 y = I$. In (4.37b), B_j are the Bernoulli numbers, with the first several given by $B_0 = 1$, $B_0 = -\frac{1}{2}$, $B_1 = \frac{1}{6}$, and $B_3 = 0$ (c.f. Bullo and Lewis [12]).

The second option is the Cayley map, $\operatorname{cay} : \mathfrak{g} \to G$. The Cayley map is defined as $\operatorname{cay} \xi = \left(I - \frac{\xi}{2}\right)^{-1} \left(I + \frac{\xi}{2}\right)$ and is valid for a general glass of quadratic groups.

Definition 4.3. *The right-trivialized derivative of the* cay *map and its inverse are given by (c.f.*

[12])

$$d\,\mathrm{cay}_x y = \left(e - \frac{x}{2}\right)^{-1} y \left(e + \frac{x}{2}\right)^{-1} \tag{4.38a}$$

$$d\,\mathrm{cay}_x^{-1} y = \left(e - \frac{x}{2}\right) y \left(e + \frac{x}{2}\right) \tag{4.38b}$$

An additional option is to use canonical coordinates of the second kind, which are based on the exponential map, but not used or considered in the present work.

4.5.4 Asymptotically Stable Closed Loop Dynamics

The notion of global stability used in this section is related to the existence of a global strict Lyapunov function. According to Bacciotti and Rosier [6], a system in the form of an ordinary differential equation

$$\dot{\boldsymbol{s}} = \boldsymbol{f}(\boldsymbol{s}) \tag{4.39}$$

such as (4.28) is Lyapunov stable at the origin if for each $\epsilon > 0$, there exists $\delta > 0$ such that for each initial state $\boldsymbol{s}_0$ with $\|\boldsymbol{s}_0\| < \delta$, the solutions $\boldsymbol{s}(t)$ with $t \in [0, \infty\ )$ satisfy

$$\|\boldsymbol{s}(t)\| < \epsilon \ \ \forall t \geq 0.$$

With the open ball of radius r about the origin defined as $B_r = \{\boldsymbol{x} \in \mathbb{R}^n : \|\boldsymbol{x}\| < r\}$, a strict smooth Lyapunov function meets the following criteria.

Definition 4.4. *A strict smooth Lyapunov function is a real map $V(\boldsymbol{x}) : B_r \to \mathbb{R}$ that satisfies*

(i) $V(\mathbf{0}) = 0$,

(ii) $V(\boldsymbol{s}) > 0$ *for* $\boldsymbol{s} \neq \mathbf{0}$,

(iii) $V(\boldsymbol{s})$ *is of class* C^1, *and*

(iv) $\nabla V(\boldsymbol{s}) \cdot \boldsymbol{f}(\boldsymbol{s}) < 0$ *except at the origin,*

so that $V(\boldsymbol{s})$ is positive definite and $\dot{V} = \nabla V(\boldsymbol{s}) \cdot \boldsymbol{f}(\boldsymbol{s})$ is negative semidefinite.

A Lyapunov function is said to be global if it is also radially unbounded, so that Definition 4.4 holds for $r \to \infty$. To develop a globally asymptotically stable controller, we will make use of Lyapunov's

second theorem, which is given in [6] as follows.

Theorem 4.1. (Second Lyapunov Theorem.) *If there exists a smooth global strict Lyapunov function, then the system* (4.39) *is globally asymptotically stable to the origin.*

In the case when $\dot{V} = \nabla V(\boldsymbol{s}) \cdot \boldsymbol{f}(\boldsymbol{s}) \leq 0$, as opposed to the strict equality rendering the Lyapunov function 'weak' as opposed to 'strict', LaSalle's Theorem yields a similar result.

Theorem 4.2. (LaSalle's Theorem.) *If there exists a smooth global weak Lyapunov function, then the system* (4.39) *is globally asymptotically stable to the origin if the only solution of* (4.39) *along with* $\dot{V}(\boldsymbol{s}) = 0$ *is* $\boldsymbol{s}(t) = \boldsymbol{0}$.

With these preliminaries in hand, define the closed loop dynamics (4.28) with desired control function $\boldsymbol{\tau}_d$ and $\boldsymbol{f}_d$ defined by

$$\boldsymbol{\tau}_d = k_R \boldsymbol{e}_R - k_\omega \boldsymbol{e}_\omega - \boldsymbol{J}\boldsymbol{\omega} \times \boldsymbol{\omega} - \boldsymbol{\tau}_{\text{ext}} + \Gamma_\vartheta(\boldsymbol{e}_R) + \frac{d}{dt}\boldsymbol{J}\,\text{Ad}_{R_e}\boldsymbol{\omega}_d \qquad \text{: desired torque} \tag{4.40a}$$

$$\boldsymbol{f}_d = -k_\varphi \boldsymbol{e}_\varphi - k_v \boldsymbol{e}_v - \boldsymbol{f}_{\text{ext}} + \boldsymbol{M}\ddot{\boldsymbol{x}}_d \qquad \text{: desired force.} \tag{4.40b}$$

where $k_R, k_\omega, k_\varphi, k_v$ are positive proportional and damping terms. The function $\Gamma_\vartheta : \mathfrak{so}(3) \to T_u^*\mathcal{U}$ depends on the choice of the retraction map ϑ used to define the orientation error. If the exponential map is used, then $\Gamma_{\exp}(\boldsymbol{e}_R) = \boldsymbol{0}$. If the Cayley map is used, then $\Gamma_{\text{cay}}(\boldsymbol{e}_R) = \frac{1}{4}k_R(\boldsymbol{e}_R \cdot \boldsymbol{e}_R)\boldsymbol{e}_R$ to accommodate for the fact that the Cayley map represents an approximation to the exponential map.

To show the global asymptotic stability to $\{\boldsymbol{e}_\varphi, \boldsymbol{e}_v, \boldsymbol{e}_R, \boldsymbol{e}_\omega\} = \boldsymbol{0}$, we use the fact that the full system of equations represents a system in cascade form. In other words, the velocity (angular or translational) can be considered as input driving the kinematic equations, whose output then drives the dynamics. Following Tsiortras [83], we show asymptotic closed loop stability by first showing that the kinematic subsystem is stabilizable–by treating the velocity errors as control-like variables–and then using these results to show the overall stability of the system.

Proposition 4.1. *The close loop kinematics in* (4.28a) *and* (4.28b) *with control functions* $\boldsymbol{e}_\omega = k_R\boldsymbol{e}_R$ *and* $\boldsymbol{e}_v = -k_\varphi \boldsymbol{e}_\varphi$ *are globally asymptotically stable to* $\{\boldsymbol{e}_R, \boldsymbol{e}_\varphi\} = \boldsymbol{0}$.

Proof. *Define the positive definite radially unbounded Lyapunov function*

$$\begin{aligned} U &= U_x(\boldsymbol{e}_\varphi) + U_R(\boldsymbol{e}_R) \\ &= \frac{1}{2}\boldsymbol{e}_R \cdot \boldsymbol{e}_R + \frac{1}{2}\boldsymbol{e}_\varphi \cdot \boldsymbol{e}_\varphi \geq 0, \end{aligned}$$

with time-derivative

$$\dot{U} = \boldsymbol{e}_R \cdot \dot{\boldsymbol{e}}_R + \boldsymbol{e}_\varphi \cdot \dot{\boldsymbol{e}}_x.$$

By differentiation and application of (4.36)*, we find that*

$$\begin{aligned}\dot{\boldsymbol{e}}_R &= \frac{d}{dt}\vartheta^{-1}(R_e)\\ &= d\vartheta^{-1}_{\boldsymbol{e}_R}\left(-\boldsymbol{\omega} + \operatorname{Ad}_{\vartheta(\boldsymbol{e}_R)}\boldsymbol{\omega}_d\right)\\ &= d\vartheta^{-1}_{\boldsymbol{e}_R}\left(-\boldsymbol{e}_\omega\right),\end{aligned}$$

for $\vartheta = \exp$*, using the property* $x \cdot (\operatorname{ad}^j_x y) = 0$*,* $j = 1, \ldots, \infty$*, we can write*

$$\begin{aligned}\dot{U} &= \boldsymbol{e}_R \cdot d\exp_{\boldsymbol{e}_R}(-\boldsymbol{e}_\omega) + \boldsymbol{e}_\varphi \cdot \dot{\boldsymbol{e}}_x\\ &= -\boldsymbol{e}_R \cdot \boldsymbol{e}_\omega + \boldsymbol{e}_\varphi \cdot \boldsymbol{e}_v\\ &= -k_R \boldsymbol{e}_R \cdot \boldsymbol{e}_R - k_\varphi \boldsymbol{e}_\varphi \cdot \boldsymbol{e}_\varphi \leq 0.\end{aligned}$$

where $k_R, k_\varphi > 0$*, and the last inequality is only equal to zero if* $\{\boldsymbol{e}_R, \boldsymbol{e}_\varphi\} = \boldsymbol{0}$*. For for* $\vartheta = \operatorname{cay}$*, we can simplify the expression* $d\operatorname{cay}^{-1}_x y = y + \frac{1}{2}\operatorname{ad}_x y - \frac{1}{4}xyx$*. Using this relationship and* $x \cdot (\operatorname{ad}^j_x y) = 0$*,* $j = 1, \ldots, \infty$*, we have*

$$\begin{aligned}\dot{U} &= \boldsymbol{e}_R \cdot d\operatorname{cay}_{\boldsymbol{e}_R}(-\boldsymbol{e}_\omega) + \boldsymbol{e}_\varphi \cdot \dot{\boldsymbol{e}}_x\\ &= \boldsymbol{e}_R \cdot \left(-\boldsymbol{e}_\omega - \frac{1}{4}(\boldsymbol{e}_\omega \cdot \boldsymbol{e}_R)\boldsymbol{e}_R\right) + \boldsymbol{e}_\varphi \cdot \boldsymbol{e}_v\\ &= -k_R\left(\boldsymbol{e}_R \cdot \boldsymbol{e}_R + \frac{1}{4}(\boldsymbol{e}_R \cdot \boldsymbol{e}_R)^2\right) - k_\varphi \boldsymbol{e}_\varphi \cdot \boldsymbol{e}_\varphi \leq 0.\end{aligned}$$

We therefore have

$$\dot{U} \leq -2\min(k_R, k_\varphi)U,$$

which implies by Theorem 4.1 that the system is in fact exponentially *stable to the origin* $\{\boldsymbol{e}_R, \boldsymbol{e}_\varphi\} = \boldsymbol{0}$*, for either choice of retraction map.* □

Corallary 4.3. *The closed loop kinematics in* (4.28a) *and* (4.28b) *with control functions* $\boldsymbol{e}_\omega = k_R \boldsymbol{e}_R$ *and* $\boldsymbol{e}_v = -k_\varphi \boldsymbol{e}_\varphi$ *are independently globally asymptotically stable to* $\boldsymbol{e}_R = \boldsymbol{0}$ *and* $\boldsymbol{e}_\varphi = \boldsymbol{0}$*.*

Proof. *Follows directly from Proposition 4.1, using the Lyapunov functions* U_R *and* U_x*.* □

With the kinematics stabilized, we use this result to stabilize the overall system.

Proposition 4.2. *The closed loop dynamics* (4.28) *with control function* $\boldsymbol{\tau}_d$ *and* $\boldsymbol{f}_d$ *defined by* (4.40) *asymptotically stabilizes to a desired trajectory defined by* $\{\boldsymbol{e}_\varphi, \boldsymbol{e}_v, \boldsymbol{e}_R, \boldsymbol{e}_\omega\} = \boldsymbol{0}$*.*

Proof. *Define the Lyapunov function*

$$V = \frac{1}{2}\left(\boldsymbol{e}_\omega^T \boldsymbol{J}\boldsymbol{e}_\omega + \boldsymbol{e}_v^T \boldsymbol{M}\boldsymbol{e}_v\right) + k_\varphi U_x(\boldsymbol{e}_\varphi) + k_R U_R(\boldsymbol{e}_R) \geq 0$$

We have, after substituting in (4.28) *and* (4.40) *and using* $\vartheta = \exp$,

$$\begin{aligned}
\dot{V} &= \boldsymbol{e}_\omega \cdot \boldsymbol{J}\dot{\boldsymbol{e}}_\omega + \boldsymbol{e}_v \cdot \boldsymbol{M}\dot{\boldsymbol{e}}_v + k_R \boldsymbol{e}_R \cdot \dot{\boldsymbol{e}}_R + k_\varphi \boldsymbol{e}_\varphi \cdot \dot{\boldsymbol{e}}_x \\
&= \boldsymbol{e}_\omega \cdot (-k_\omega \boldsymbol{e}_\omega + k_R \boldsymbol{e}_R) + \boldsymbol{e}_v \cdot (-k_v \boldsymbol{e}_v - k_\varphi \boldsymbol{e}_\varphi) + k_R \boldsymbol{e}_R \cdot \dot{\boldsymbol{e}}_R + k_\varphi \boldsymbol{e}_\varphi \cdot \dot{\boldsymbol{e}}_x \\
&= -k_\omega \boldsymbol{e}_\omega \cdot \boldsymbol{e}_\omega - k_v \boldsymbol{e}_v \cdot \boldsymbol{e}_v + k_R \boldsymbol{e}_R \cdot (\dot{\boldsymbol{e}}_R + \boldsymbol{e}_\omega) \\
&= -k_\omega \boldsymbol{e}_\omega \cdot \boldsymbol{e}_\omega - k_v \boldsymbol{e}_v \cdot \boldsymbol{e}_v \leq 0.
\end{aligned}$$

Using $\vartheta = \mathrm{cay}$,

$$\begin{aligned}
\dot{V} &= \boldsymbol{e}_\omega \cdot \boldsymbol{J}\dot{\boldsymbol{e}}_\omega + \boldsymbol{e}_v \cdot \boldsymbol{M}\dot{\boldsymbol{e}}_v + k_R \boldsymbol{e}_R \cdot \dot{\boldsymbol{e}}_R + k_\varphi \boldsymbol{e}_\varphi \cdot \dot{\boldsymbol{e}}_x \\
&= \boldsymbol{e}_\omega \cdot (-k_\omega \boldsymbol{e}_\omega + k_R \boldsymbol{e}_R + \Gamma_{\mathrm{cay}}(\boldsymbol{e}_R)) + \boldsymbol{e}_v \cdot (-k_v \boldsymbol{e}_v - k_\varphi \boldsymbol{e}_\varphi) + k_R \boldsymbol{e}_R \cdot \dot{\boldsymbol{e}}_R + k_\varphi \boldsymbol{e}_\varphi \cdot \dot{\boldsymbol{e}}_x \\
&= -k_\omega \boldsymbol{e}_\omega \cdot \boldsymbol{e}_\omega - k_v \boldsymbol{e}_v \cdot \boldsymbol{e}_v + k_R \boldsymbol{e}_R \cdot \left(-\boldsymbol{e}_w + \boldsymbol{e}_\omega - \frac{1}{4}(\boldsymbol{e}_R \cdot \boldsymbol{e}_\omega)\boldsymbol{e}_R\right) + \boldsymbol{e}_\omega \cdot \Gamma_{\mathrm{cay}}(\boldsymbol{e}_R) \\
&= -k_\omega \boldsymbol{e}_\omega \cdot \boldsymbol{e}_\omega - k_v \boldsymbol{e}_v \cdot \boldsymbol{e}_v \leq 0.
\end{aligned}$$

The proposition holds since for either choice of retraction map since $V \geq 0$ *and* $\dot{V} \leq 0$ *with equality only when all errors are zero. This can be seen by substituting* (4.40) *into* (4.28)*; at* $\dot{V} \equiv 0$, $\boldsymbol{e}_\omega = \dot{\boldsymbol{e}}_\omega \equiv \boldsymbol{0} \Rightarrow \boldsymbol{e}_R \equiv \boldsymbol{0}$ *and* $\boldsymbol{e}_v = \dot{\boldsymbol{e}}_v \equiv \boldsymbol{0} \Rightarrow \boldsymbol{e}_\varphi \equiv \boldsymbol{0}$*. Thus, by LaSalle's theorem, the system is globally asymptotically stable at the origin.* □

4.5.5 Docking Trajectory Design

The basic requirements of the docking trajectories are outlined in Section 4.5.2. One option is to pre-compute an optimal trajectory–or a family of optimal trajectories–and then use a method along the lines of Kobilarov [43] to generate a trajectory. A simpler approach is to explicitly define a family of smooth curves that satisfy the requirements laid out in Section 4.5.2 for any initial state and guide the spacecraft towards the final docked configuration. While the examples in this section assume full 6 DoF actuation with no limits on the magnitude of control forces and torques, in reality the system is likely to be underactuated. In future applications for which the dynamics are not fully controlled, we would ideally like to track a trajectory for which the natural dynamics of the uncontrolled degrees of freedom keep the spacecraft close to the desired trajectory.

From this perspective, defining a *family* of curves is useful so that the desired trajectory can be dynamically updated based on the current state of the spacecraft, and a new trajectory computed if the spacecraft has drifted too far from the original trajectory. In addition, the proposed cup-and-cone docking port design (see Section 3.7.2) readily allows for slightly oblique final approaches with the feedback controller *still active* when the docking ports are in contact. In fact, in the event that there is some translational mis-alignment between matching ports, there needs to be a corresponding angular misalignment, or obliqueness in the approach, to make use of the port geometry and ensure a properly aligned final docked configuration.

To this end, we define trajectories using magnetic field lines emanating from virtual point dipoles at the center of the receiving spacecraft and aligned with the docking ports, $\boldsymbol{b}_v(t)$, given by

$$\boldsymbol{b}_v := 3\bar{\boldsymbol{d}}(\boldsymbol{c}_0 \cdot \bar{\boldsymbol{d}}) - \boldsymbol{c}_0 \tag{4.41}$$

in which $\boldsymbol{d}$ is the distance from the dipole moment to a point is space, $\bar{\boldsymbol{d}} = \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}$, and $\boldsymbol{c}_0$ is the orientation of the moment.

In the typical representation of a dipole field, Equation (4.41) is premultiplied by the term $\frac{m_0\mu_0}{4\pi\|\boldsymbol{d}\|^3}$, in which m is the magnitude of the moment and $\mu_0 = 4\pi 10e - 7$, but these terms are immaterial to the use of the virtual field lines to define docking trajectories. In this application of (4.41), $\boldsymbol{d}$ is the distance between the centers of mass of the docking spacecrafts, and $\boldsymbol{c}_0$ is the alignment of the moment of the virtual dipole on the receiving spacecraft, which without loss of generality is assumed to be aligned with the docking ports along $\boldsymbol{d}_{1,d}$, i.e. the $x-$direction in the body frame. Several example trajectories are shown in Figure 4.30.

At each point along the trajectory, we require the spacecraft to move with speed $\eta\|\boldsymbol{b}_v\|$ along the virtual magnetic field line direction $-\bar{\boldsymbol{b}}_v$ (note that $\|\boldsymbol{b}_v\| \in [1, 2]$, so the maximum speed is given by 2η). Thus, $\boldsymbol{v}_d$ is given by

$$\boldsymbol{v}_d(t) = -\eta \boldsymbol{b}_v.$$

We set $\boldsymbol{d}_{1,d} = -\bar{\boldsymbol{b}_v}$ so that the desired orientation at each position along the trajectory is:

$$\boldsymbol{R}_d(t) = \left[-\bar{\boldsymbol{b}_v} \mid \bar{\boldsymbol{a}}_2 \mid -\bar{\boldsymbol{b}_v} \times \bar{\boldsymbol{a}}_2\right].$$

For this family of trajectories $\bar{\boldsymbol{b}}_v \to \bar{\boldsymbol{d}}$ as $\boldsymbol{\varphi} \to \tilde{\boldsymbol{\varphi}}_0$. This means that the desired position is simply

the next position along the virtual field line, giving

$$\boldsymbol{\varphi}_d(t) = \boldsymbol{\varphi}_0 + \int \boldsymbol{v}_d(t)dt.$$

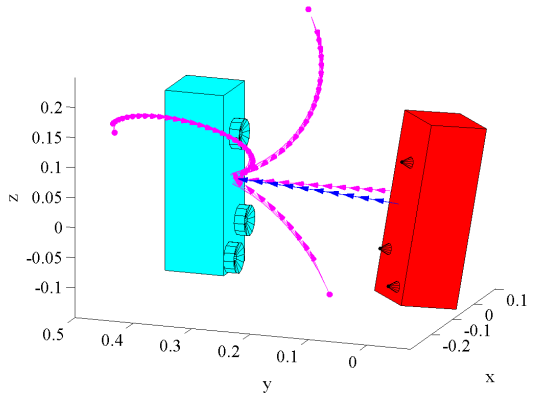


Figure 4.30: Representation of CubeSat geometry for the AAReST Mission, along with example trajectories emanating from various initial configurations of the male CubeSat calculated from virtual magnetic field lines. The trajectory followed for the example configuration is shown in blue. Other examples are shown in magenta.

4.5.6 Example: Autonomous Spacecraft Docking

This section demonstrates the feedback stabilized controller in Section 4.5 using Explicit Newmark for time-integration and the potential-based approach to model elastic collisions, thus taking advantage of the cup-and-cone docking port design to bring the CubeSats into final alignment. In this example, one spacecraft (the 'female' craft) is assumed to be stationary, and the control solution (either feedback or optimal) is calculated for the 'male' craft, as shown in Figure 4.30. Both non-convex spacecrafts are divided into convex parts, with the cup geometry modeled using 8 separate polyhedra, and the cones correspondingly modeled using faceted prisms with 8 facets, and the components of the kinematic docking system also included. This leads to a total of 7 convex sub-bodies for the male craft, and 41 for the female craft. The dimensions of the craft are 0.1m in the short directions, and 0.3m in the long direction–the size of a 3 unit or '3u' CubeSat.

The initial translational and angular velocity of the male craft are $\dot{\varphi}^0 = [-1e{-}3,\ 5e{-}3,\ 0]^{\text{m}/\text{s}}$ and $\boldsymbol{\omega}^0 = [-1e{-}3,\ 5e{-}3,\ 0]^{\text{m}/\text{s}}$, respectively. Without actuation, these initial conditions would cause the moving craft to pass the stationary one and continue moving away, as shown in Figure 4.31.

The parameter η is chosen as $\eta = 5e{-}2$, so that in the feedback stabilized scheme the maximum

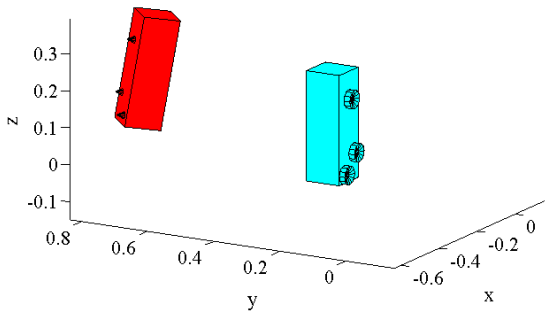
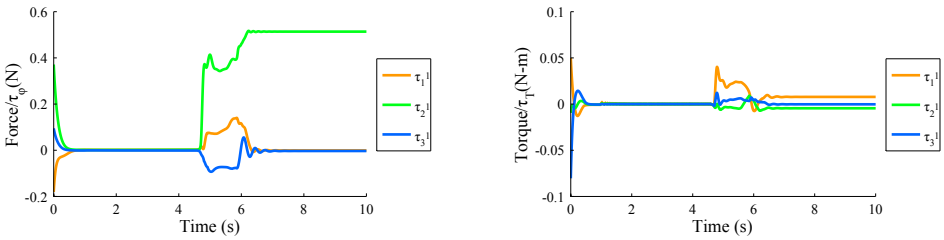


Figure 4.31: Configurations after 10s with initial condition in 4.33a if there is no actuation to facilitate docking.

translational speed of the craft is $10^{\text{cm}}/_{\text{s}}$. The feedback gains $k_R, k_\omega, k_\varphi, k_v$ are given by

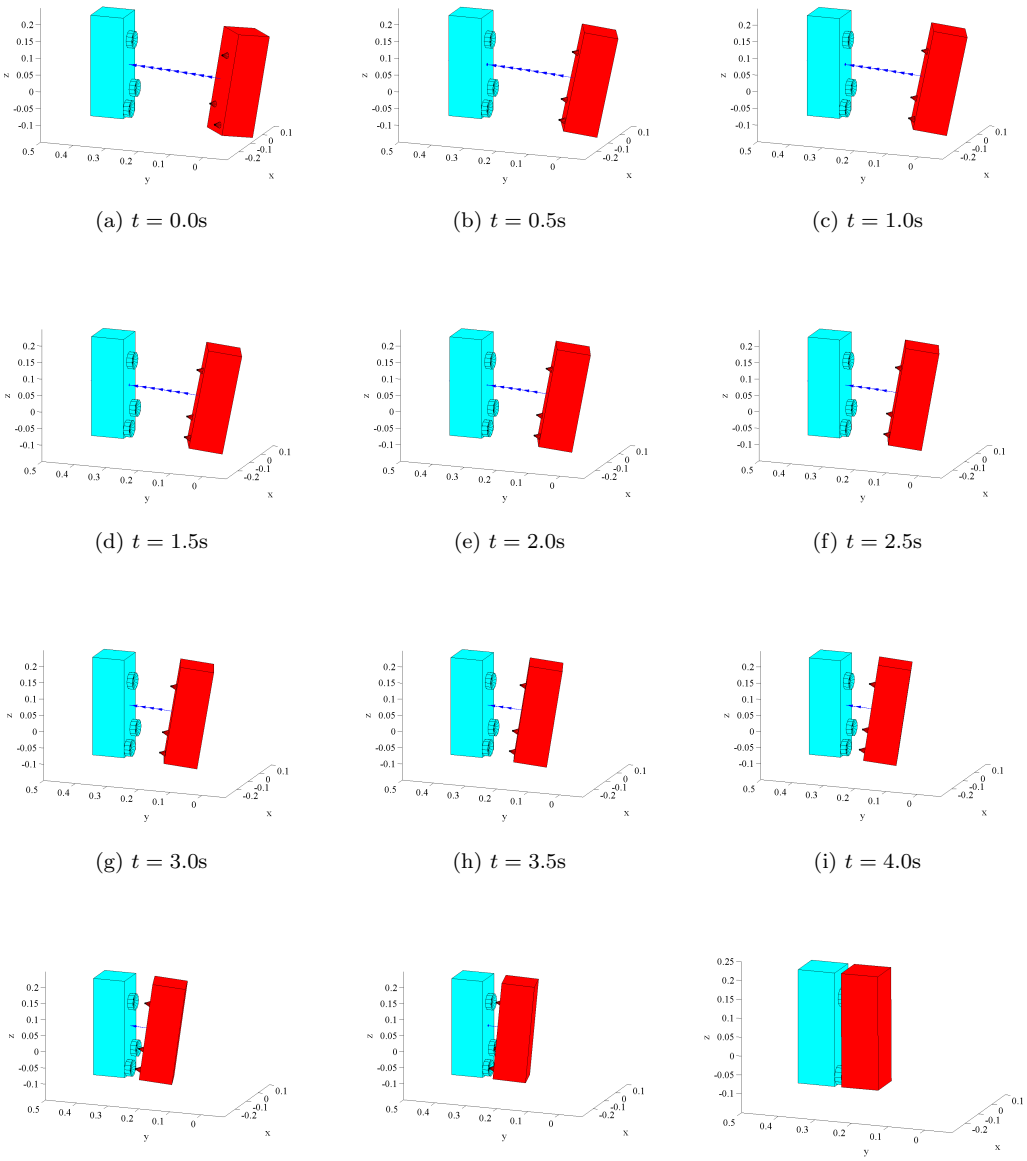
$$k_R = 1.0,$$
$$k_\omega = 0.3,$$
$$k_\varphi = 1.2, \text{ and}$$
$$k_v = 9.6,$$

and the total mass of the craft is $M_\varphi = 1.2$ kg. The center of mass and inertial properties are approximated as being equivalent to a constant density rectangular prism. The resulting control forces and torques for a $t_K = 10$s maneuver are shown in Figure 4.32, and time laps images of the motion can be found in Figure 4.33. The time step for the Newmark integrator is $h = 1e{-}3$s. After the initial collision in which the pins enter the cones but the bodies are not yet aligned, the continued application of the same control law causes the bodies to come into alignment at about $t \sim 6.2$s. The change in the quality of the control forces and torques are due to the fact that the controller is unaware that the contact is taking place (i. e., contact forces and torques are not included in $\boldsymbol{f}_{ext}$ and $\boldsymbol{\tau}_{ext}$). The total control effort through 6.2s, calculated in the same way as the DMOCC problems as the sum of control forces and torques squared times the time step is $2.5781e{-}1$N·s.



(a) Feedback control, $\boldsymbol{f}_c \equiv \boldsymbol{\tau}_\varphi$ (b) Feedback Control, $\boldsymbol{\tau}_c \equiv \boldsymbol{\tau}_T$

Figure 4.32: Evolution of feedback stabilized control forces and torques.



(a) $t = 0.0$s (b) $t = 0.5$s (c) $t = 1.0$s

(d) $t = 1.5$s (e) $t = 2.0$s (f) $t = 2.5$s

(g) $t = 3.0$s (h) $t = 3.5$s (i) $t = 4.0$s

(j) $t = 4.5$s (k) $t = 5.0$s (l) $t = 6.5$s

Figure 4.33: Stabilization to a docking trajectory and final docked configuration using virtual magnetic field lines to generate the trajectory. The trajectory is sketched in blue.

4.6 Conclusion

In this chapter, a series of new control strategies have been developed for the design of maneuvers in which collisions between non-smooth and possibly non-convex bodies need to either be avoided, explicitly included in the trajectory, or some combination of the two. The SSH LP is used extensively to enable the design of 'tight' maneuvers between non-smooth and non-convex bodies which may start or end in contact configurations, and must pass extremely close to contact configurations during the trajectory. It is also shown that the inverse dynamics problem leads to a feasible, but certainly far from optimal in the sense of control effort, initial guess. Examples including the solution to the knot-burr parlor puzzle and the control of an articulated manipulator demonstrated the broad applicability of the method in finding solutions for complicated assembly processes.

Two alternatives are presented for planning maneuvers in which contact interactions cannot–or it is desired that contact interactions are not–avoided. In the first method, the unilateral contact potential is introduced into the projected discrete equations of motion which constrain the optimal control problem. While this is straightforward from an implementation standpoint, the meaning of control forces applied when two rigid bodies are overlapping is suspect. A more accurate set of constraints for the discrete optimal control problem can be formulated by adapting the momentum reflection technique developed for the DCR-DCM algorithm in the previous chapter. In the momentum reflection formulation, the discrete trajectory is pre-seeded with a designated set of contact nodes, and the base time step within each subtrajectory (the intervals between the contact nodes) is allowed to vary so that the physical collision time is left to the optimizer. It is shown that for a simple example, the choice of the contact node set does not significantly effect the physical contact time or the evolution of the generalized forces in the optimal solution. The method is limited, however, by the need to at least designate the *number* of collisions which take place in the maneuver.

Finally, the problem of robust CubeSat docking is revisited, and motivates the design of a globally asymptotically feedback stabilized control law capable of following a desired trajectory which, if properly designed, leads to successful spacecraft docking. The controller is tested for a family of trajectories generated from virtual magnetic field lines and, by example, the is shown to successfully lead to the desired final alignment even if the ports are not aligned when they first come into contact for the kinematic docking port design discussed in Section 3.7.2.

5 Conclusion and Future Directions

Motivated by problems inherent in modeling self-assembling systems, this thesis has presented advances in the areas of collision detection and integration for non-smooth bodies, and in optimal control for problems involving collisions or tightly constrained maneuvers involving non-smooth bodies. In Chapter 2, the SSH LP was introduced as a robust sub-differentiable collision detection algorithm, with the added benefits that the subgradient is both normal to a contact surface and local to features of the body involved in the collision. Furthermore, exactly one interpenetration condition needs to be checked for each pair of convex bodies (or sub-bodies), as opposed to other approaches in which the bodies must be decomposed further into triangles or line segments. While the nature of linear finite elements and self-assembling macroscale mechanical (most of which in literature are composed of non-smooth parts) systems motivated the in-depth treatment of the SSH LP, further analysis and application of the method to general convex bodies described by implicit surfaces should be pursued. Fruitful extensions of the method could also be found in machine perception where it could be used in conjunction with, e.g. Peters et al. [72], to characterize the geometry and topology of objects in a robots environment. Finally, because in addition to giving overlap information, the SSH LP also provides exact distance information, the extension of the method to modeling surface interactions between non-smooth bodies, and possibly even its direct application in a discrete or granular element method (c. f. Andrade et al. [4], Radjai and Richefeu [74]), should also be explored.

The utility of the SSH LP in formulating numerical integration schemes with varying levels of accuracy and efficiency is demonstrated in Chapter 3. In the most straightforward scheme from an implementation perspective, the SSH LP can be used directly as unilateral potential which is a smooth approximation to the indicator function. This leads to inaccuracy in the sense that bodies in the system must be allowed to achieve inadmissible states and overlap by a small amount in order for the method to work. While it cannot be completely eliminated, the amount of overlap is tunable via the contact potential parameter, C, although changes in this parameter must frequently be accompanied by modifications to the base time-step, h, in order to prevent numerical instabilities.

In future work, it is worth pursuing the combination of the SSH LP-based contact potential with time-adaptive or asynchronous integrators (c.f. Lew et al. [47]) to enable more efficient models of self-assembling systems with a minimal loss of accuracy using this method.

A less efficient but more exact method of collision integration is found by introducing the exact collision time (and corresponding configuration) as an additional variable in the constrained discrete variational setting. This leads to a set of discrete impact equations which are analogous to the jump conditions on energy and momenta in the continuous setting. While useful due to the exact conservation of discrete energy and momenta through each collision, this method become prohibitively expensive for self-assembling systems because each in a series of closely spaced (in time) collisions must be exactly resolved. This problem is compounded when polyhedral bodies are present in the system because of the potential of collisions between these bodies to induce relative rotation between the bodies. As such, the series of collisions to be resolved may involve multiple collisions in close succession between the *same* pair of bodies, which further the resolution of an overlap-free post-collision configuration.

A compromise between accuracy and efficiency in the setting of (an approximation to) a discrete constrained variational integrator is found in the DCR-DCM algorithm. In this algorithm, collision times are never resolved exactly. Rather, if overlap is detected during a time step, a feasible collision configuration is determined by recourse to a closest point project operation. With the assumed collision configuration in hand, explicit momentum updates are formulated which can be made to include inelastic or frictional collisions in a straightforward way. By the use of this algorithm, one accepts that small changes–which can be exactly tracked to verify the expected conservation properties of the integrator–will be made to the conserved components of the total angular momentum and total energy (if potentials are present). As in the potential-based approach, these small changes can be minimized by decreasing the base time step of the algorithm, but can never be completely eliminated. However, gains might still be made in simulations of very large self-assembling systems by introduction time adaption into this approach. In contrast to the potential-based approach, the method itself places no upper limit on the time-step, aside from that needed to maintain numerical stability in the integration of the overall system.

In Chapter 4, the robust collision detection developed in Chapter 2 and the discrete variational collision integrators in Chapter 3 are used to develop and/or test a series of control algorithms in the DMOCC and feedback-stabilized settings. The first set of optimal control problems illustrate the use of the SSH LP as a path constraint in the DMOCC setting to enable the design of complicated assembly maneuvers between non-smooth and non-convex bodies or kinematic chains of bodies. The resulting maneuvers allow the bodies to move seamlessly in ‘tight’ configurations in which

collisions are only very narrowly avoided. In the sequel, a family of optimal control problems in which collisions cannot or should not be avoided is formulated, taking advantage of the momentum reflection techniques of the DCR-DCM algorithm. Finally, a globally stable feedback stabilized controller for trajectory tracking which employs a minimal parametrization of the error terms is developed. The feedback controlled results using a family of trajectories based on a virtual dipole field are tested with the real geometry of the AAReST mission CubeSats. In future work, the baseline optimal feedback controller will be extended to robustly treat underactuated and highly non-linear actuation systems such as electromagnetically controlled spacecraft docking.

In addition to improving the efficiency of the collision integrators developed for non-smooth contact, the techniques here could be developed–perhaps by integration with an existing orbital dynamics code–into a robust modular numerical testbed for design and certification of multi-agent, autonomous systems, including self-assembling space systems. Furthermore, the DMOCC problems and formulations presented herein suggest the need for better ways to generate an initial guess, and also the need for a more general formulation of the optimal control problem in which contact plays a role in the dynamics. To this end, there is work to be done in developing new methods in numerical optimization for the kind of non-smooth non-convex problems encountered in the aforementioned design and control problems, for example in mixed integer non-linear programming (c.f. Exler et al. [24]). Finally, the aforementioned testbed and optimal control formulations could be used in conjunction with the certification methodology developed in Owhadi et al. [67] to design quantifiably robust self-assembling space systems.

To conclude, this thesis has developed robust tools which enable the direct numerical simulation and optimization of the complex dynamics of self-assembling systems with the goal of enabling the design of next-generation self-assembling systems for space applications. In addition to providing the basis of a robust numerical testbed for self-assembling systems and advancing the state of the art in several areas of analysis and simulation, the developments herein suggest extensions to and improvements to be made in related fields, and further expansion of the breadth and depth of contact, control, and optimization problems at several scales.

A Articulated Manipulator

This appendix details the mapping $\boldsymbol{C}^C$, null space matrix, $\boldsymbol{P}_{ext}^C$, and the explicit update in terms of the local nodal reparametrization used to solve the DMOCC problem with non-smooth collision avoidance for the articulated manipulator in Section 4.3. Recall that the articulated manipulator is modeled as a kinematic chain in Figure 4.11, which is reproduced below.

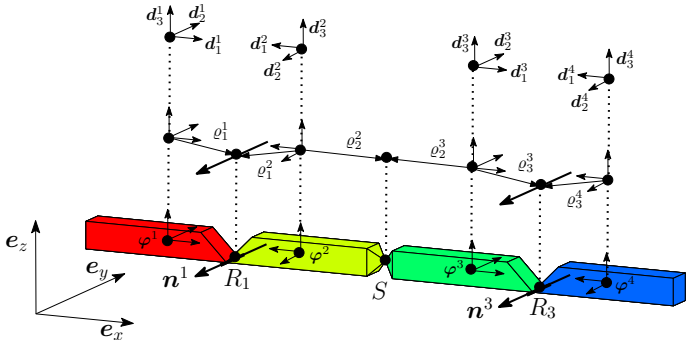


Figure A.1: Schematic diagram of the assembled kinematic chain of four polyhedral bodies used to model an articulated manipulator. Joint locations and kinematic constraints are induced naturally from the geometry of the bodies. Reproduction of Figure 4.11.

A.0.1 Redundant Forces

For a kinematic pair J, the mapping $\boldsymbol{B}^T$ has the form

$$\boldsymbol{B}^T(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{P}_{int}(\boldsymbol{q}^1) & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{P}_{int}(\boldsymbol{q}^2) \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{1}{2}\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \frac{1}{2}\boldsymbol{I} \end{bmatrix} \cdot \begin{bmatrix} \boldsymbol{C}_{rb}(\boldsymbol{q}^1) & \boldsymbol{C}^{1,J}(\boldsymbol{q}) \\ \boldsymbol{0} & \boldsymbol{C}^{2,J}(\boldsymbol{q}) \end{bmatrix}.$$

What remains is to determine $\boldsymbol{C}^J$ for each pair of bodies, and finally to link the chain together to form $\boldsymbol{C}^C$. To this end, the results of Leyendecker et al. [52] are summarized below.

For the spherical pair, $\boldsymbol{C}^{1,S}$, $\boldsymbol{C}^{2,S} \in \mathbb{R}^{6\times 3}$ are given by

$$\boldsymbol{C}^{1,S}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{I} \end{bmatrix}, \qquad \boldsymbol{C}^{2,S}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{I} \end{bmatrix}. \tag{A.1}$$

Likewise, for the revolute pair, $\boldsymbol{C}^{1,R}$, $\boldsymbol{C}^{2,R} \in \mathbb{R}^{6\times 1}$ are given by

$$\boldsymbol{C}^{1,R}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{0} \\ -\boldsymbol{n}^1 \end{bmatrix}, \qquad \boldsymbol{C}^{2,R}(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{n}^1 \end{bmatrix}. \tag{A.2}$$

In (A.1) and (A.2), the formulation of $\boldsymbol{C}^C$ is determined by the kinematic constraints imposed by the joints. By combining (A.1) and (A.2), the redundant forces for the complete chain in this example (see Figure 4.11) can be recovered using

$$\boldsymbol{C}^C(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} \\ \widehat{\boldsymbol{\varrho}}_{rb} & \boldsymbol{I} & -\boldsymbol{n}^1 & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{n}^1 & -\boldsymbol{I} & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} & \boldsymbol{I} & -\boldsymbol{n}^3 \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 1} \\ \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 3} & \boldsymbol{0}_{3\times 1} & \boldsymbol{0}_{3\times 3} & \boldsymbol{n}^3 \end{bmatrix} \in \mathbb{R}^{48\times 11}. \tag{A.3}$$

A.0.2 Null Space Matrix

The key to determining the null space matrix for the full system is to relate the joint velocities to the (redundant) rigid body twists, which can be found by successive backsubstitution of (3.27), (3.31), and (3.32). In this example,

$$\boldsymbol{\nu}^C = \begin{bmatrix} \dot{\varphi}^1 \\ \boldsymbol{\omega}^1 \\ \dot{\theta}^2 \\ \boldsymbol{\omega}^3 \\ \dot{\theta}^4 \end{bmatrix} \in \mathbb{R}^{11}, \tag{A.4}$$

so that the superscripts on the joint velocities are associated with the second body in a particular linkage. To facilitate this process, the following notation will be used

$$\begin{aligned}
\boldsymbol{I}^{11} &= \boldsymbol{I} - \boldsymbol{n}^1 \otimes \boldsymbol{n}^1 \\
\boldsymbol{N}^{11} &= \boldsymbol{n}^1 \otimes \boldsymbol{n}^1 \\
\boldsymbol{I}^{33} &= \boldsymbol{n}^3 \otimes \boldsymbol{n}^3 \\
\boldsymbol{N}^{33} &= \boldsymbol{n}^3 \otimes \boldsymbol{n}^3.
\end{aligned}$$

For the first joint, the revolute joint R_1, for bodies 1 and 2 is found in Section 3.2.4, i.e.,

$$\boldsymbol{\omega}^2 = \boldsymbol{I}^{11} \cdot \boldsymbol{\omega}^1 + \dot{\theta}^2 \boldsymbol{n}^1 \tag{A.5a}$$

$$\dot{\boldsymbol{\varphi}}^2 = \dot{\boldsymbol{\varphi}}^1 - \left[\widehat{\boldsymbol{\varrho}_1^1 - \boldsymbol{\varrho}_1^2} - \widehat{\boldsymbol{\varrho}_1^2}\boldsymbol{N}^{11}\right] \cdot \boldsymbol{\omega}^1 + \left(\boldsymbol{\varrho}_1^2 \times \boldsymbol{n}^1\right) \dot{\theta}^2. \tag{A.5b}$$

The spherical linkage decouples the angular velocities of bodies 2 and 3, however the translational velocities are related by

$$\begin{aligned}
\dot{\boldsymbol{\varphi}}^3 &= \dot{\boldsymbol{\varphi}}^2 + \boldsymbol{\omega}^2 \times \boldsymbol{\varrho}_2^2 - \boldsymbol{\omega}^3 \times \varrho_2^3 \\
&= \dot{\boldsymbol{\varphi}}^1 - \left[\widehat{\boldsymbol{\varrho}_1^1 - \boldsymbol{\varrho}_1^2} - \widehat{\boldsymbol{\varrho}_1^2}\boldsymbol{N}^{11}\right] \cdot \boldsymbol{\omega}^1 + \left(\boldsymbol{\varrho}_1^2 \times \boldsymbol{n}^1\right) \dot{\theta}^2 + \left(\boldsymbol{I}^{11} \cdot \boldsymbol{\omega}^1 + \dot{\theta}^2 \boldsymbol{n}^1\right) \times \boldsymbol{\varrho}_2^2 - \boldsymbol{\omega}^3 \times \boldsymbol{\varrho}_2^3 \\
&= \dot{\boldsymbol{\varphi}}^1 - \left[\widehat{\boldsymbol{\varrho}_1^1 - \boldsymbol{\varrho}_1^2} - \widehat{\boldsymbol{\varrho}_1^2}\boldsymbol{N}^{11} - \widehat{\boldsymbol{\varrho}_2^2}\boldsymbol{I}^{11}\right] \cdot \boldsymbol{\omega}^1 + \left(\boldsymbol{\varrho}_1^2 \times \boldsymbol{n}^1\right) + \boldsymbol{n}^1 \times \boldsymbol{\varrho}_2^2)\dot{\theta}^2 - \boldsymbol{\omega}^3 \times \boldsymbol{\varrho}_2^3.
\end{aligned} \tag{A.6}$$

For the final revolute joint, R_3, between bodies 3 and 4, the relation ship between the twist of body 4 and the joint velocities is

$$\boldsymbol{\omega}^4 = \boldsymbol{I}^{33} \cdot \boldsymbol{\omega}^3 + \dot{\theta}^4 \boldsymbol{n}^3 \tag{A.7a}$$

$$\begin{aligned}
\dot{\boldsymbol{\varphi}}^4 &= \dot{\boldsymbol{\varphi}}^3 - \left[\widehat{\boldsymbol{\varrho}_3^3 - \boldsymbol{\varrho}_3^4} - \widehat{\boldsymbol{\varrho}_3^4}\boldsymbol{N}^{33}\right] \cdot \boldsymbol{\omega}^3 + \left(\boldsymbol{\varrho}_3^4 \times \boldsymbol{n}^3\right) \dot{\theta}^4 \\
&= \dot{\boldsymbol{\varphi}}^1 - \left[\widehat{\boldsymbol{\varrho}_1^1 - \boldsymbol{\varrho}_1^2} - \widehat{\boldsymbol{\varrho}_1^2}\boldsymbol{N}^{11} - \widehat{\boldsymbol{\varrho}_2^2}\boldsymbol{I}^{11}\right] \cdot \boldsymbol{\omega}^1 + \left(\boldsymbol{\varrho}_1^2 \times \boldsymbol{n}^1 + \boldsymbol{n}^1 \times \boldsymbol{\varrho}_2^2\right) \dot{\theta}^2 - \boldsymbol{\omega}^3 \times \boldsymbol{\varrho}_2^3 \\
&\qquad - \left[\widehat{\boldsymbol{\varrho}_3^3 - \boldsymbol{\varrho}_3^4} - \widehat{\boldsymbol{\varrho}_3^4}\boldsymbol{N}^{33}\right] \cdot \boldsymbol{\omega}^3 + \left(\boldsymbol{\varrho}_3^4 \times \boldsymbol{n}^3\right) \dot{\theta}^4 \qquad \text{(A.7b)} \\
&= \dot{\boldsymbol{\varphi}}^1 - \left[\widehat{\boldsymbol{\varrho}_1^1 - \boldsymbol{\varrho}_1^2} - \widehat{\boldsymbol{\varrho}_1^2}\boldsymbol{N}^{11} - \widehat{\boldsymbol{\varrho}_2^2}\boldsymbol{I}^{11}\right] \cdot \boldsymbol{\omega}^1 + \left(\boldsymbol{\varrho}_1^2 \times \boldsymbol{n}^1 + \boldsymbol{n}^1 \times \boldsymbol{\varrho}_2^2\right) \dot{\theta}^2 \\
&\qquad - \left[\widehat{\boldsymbol{\varrho}_3^3 - \boldsymbol{\varrho}_3^4} - \widehat{\boldsymbol{\varrho}_3^4}\boldsymbol{N}^{33} - \widehat{\boldsymbol{\varrho}_2^3}\right] \cdot \boldsymbol{\omega}^3 + \left(\boldsymbol{\varrho}_3^4 \times \boldsymbol{n}^3\right) \dot{\theta}^4.
\end{aligned}$$

To shorten notation, take

$$\begin{aligned}
\boldsymbol{A}_1^2 &= -\left[\widehat{\boldsymbol{\varrho}_1^1 - \boldsymbol{\varrho}_1^2} - \widehat{\boldsymbol{\varrho}_1^2}\boldsymbol{N}^{11}\right] \in \mathbb{R}^{3\times3}, \\
\boldsymbol{A}_2^3 &= -\left[\widehat{\boldsymbol{\varrho}_1^1 - \boldsymbol{\varrho}_1^2} - \widehat{\boldsymbol{\varrho}_1^2}\boldsymbol{N}^{11} - \widehat{\boldsymbol{\varrho}_2^2}\boldsymbol{I}^{11}\right] \in \mathbb{R}^{3\times3}, \\
\boldsymbol{A}_3^4 &= -\left[\widehat{\boldsymbol{\varrho}_3^3 - \boldsymbol{\varrho}_3^4} - \widehat{\boldsymbol{\varrho}_3^4}\boldsymbol{N}^{33} - \widehat{\boldsymbol{\varrho}_2^3}\right] \in \mathbb{R}^{3\times3}.
\end{aligned}$$

The null space matrix $\boldsymbol{P}_{ext}^C$ for the kinematic chain sketched in Figure 4.11 is then given by

$$\boldsymbol{P}_{ext}^C(\boldsymbol{q}) = \begin{bmatrix}
\boldsymbol{I}_{3\times3} & \boldsymbol{A}_1^2 & \boldsymbol{\varrho}_1^2 \times \boldsymbol{n}^1 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{I}^{11} & \boldsymbol{n}^1 & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} \\
\boldsymbol{I}_{3\times3} & \boldsymbol{A}_2^3 & \left(\boldsymbol{\varrho}_1^2 - \boldsymbol{\varrho}_2^2\right) \times \boldsymbol{n}^1 & \widehat{\boldsymbol{\varrho}_2^3} & \boldsymbol{0}_{3\times1} \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}_{3\times3} & \boldsymbol{0}_{3\times1} \\
\boldsymbol{I}_{3\times3} & \boldsymbol{A}_2^3 & \left(\boldsymbol{\varrho}_1^2 - \boldsymbol{\varrho}_2^2\right) \times \boldsymbol{n}^1 & \boldsymbol{A}_3^4 & \boldsymbol{\varrho}_3^4 \times \boldsymbol{n}^3 \\
\boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times3} & \boldsymbol{0}_{3\times1} & \boldsymbol{I}^{33} & \boldsymbol{n}^3
\end{bmatrix} \in \mathbb{R}^{18\times11}. \tag{A.8}$$

Accounting for the linkages, the $\boldsymbol{P}_{int}^C$ is given by

$$\boldsymbol{P}_{int}^C(\boldsymbol{q}) = \begin{bmatrix}
\boldsymbol{P}_{int}^2 & \boldsymbol{0}_{12\times6} & \boldsymbol{0}_{12\times6} \\
\boldsymbol{0}_{12\times6} & \boldsymbol{P}_{int}^3 & \boldsymbol{0}_{12\times6} \\
\boldsymbol{0}_{12\times6} & \boldsymbol{0}_{12\times6} & \boldsymbol{P}_{int}^4
\end{bmatrix} \in \mathbb{R}^{36\times18} \tag{A.9}$$

and the complete null space matrix for the chain $\boldsymbol{P}^C$ is then

$$\boldsymbol{P}^C(\boldsymbol{q}) = \begin{bmatrix} \boldsymbol{P}^1_{int}(\boldsymbol{q}) & \boldsymbol{0}_{12\times 5} \\ \boldsymbol{P}^C_{int}(\boldsymbol{q}) \cdot \boldsymbol{P}^C_{ext}(\boldsymbol{q}) \end{bmatrix} \in \mathbb{R}^{48\times 11}.$$

A.0.3 Local Reparametrization and Explicit Update.

The explicit update in terms of the generalized degrees of freedom

$$\boldsymbol{u}^C = \begin{bmatrix} \boldsymbol{u}^1_\varphi \\ \boldsymbol{u}^1_T \\ d\theta^2 \\ \boldsymbol{u}^3_S \\ d\theta^4 \end{bmatrix} \in \mathbb{R}^{11} \tag{A.10}$$

can be deduced in an analogous way to the complete null space matrix for the chain. To begin, define the following incremental rotation matrices, taken from (3.24), (3.26), and (3.30)

$$d\boldsymbol{R}_1 = \exp(\widehat{\boldsymbol{u}^1_{T,k}}) \tag{A.11a}$$
$$d\boldsymbol{R}_2 = \exp(-\widehat{\boldsymbol{N}^{11}\boldsymbol{u}^1_{T,k}}) \exp(\widehat{d\theta^2 \boldsymbol{n}^1}) \tag{A.11b}$$
$$d\boldsymbol{R}_3 = \exp(\widehat{\boldsymbol{u}^3_{S,k}}) \tag{A.11c}$$
$$d\boldsymbol{R}_4 = \exp(-\widehat{\boldsymbol{N}^{33}\boldsymbol{u}^3_{S,k}}) \exp(\widehat{d\theta^4 \boldsymbol{n}^3}). \tag{A.11d}$$

The updates to the center of mass locations of each body in the chain, $\boldsymbol{\varphi}^j$, are given by

$$\boldsymbol{\varphi}^1_k = \boldsymbol{\varphi}^1_{k-1} + \boldsymbol{u}_{\varphi,k} \tag{A.12a}$$
$$\boldsymbol{\varphi}^2_k = \boldsymbol{\varphi}^1_{k-1} + d\boldsymbol{R}_1 \boldsymbol{\varrho}^1_{1,k-1} - d\boldsymbol{R}_1 d\boldsymbol{R}_2 \boldsymbol{\varrho}^2_{1,k-1} \tag{A.12b}$$
$$\boldsymbol{\varphi}^3_k = \boldsymbol{\varphi}^2_{k-1} + d\boldsymbol{R}_1 d\boldsymbol{R}_2 \boldsymbol{\varrho}^2_{2,k-1} - d\boldsymbol{R}_3 \boldsymbol{\varrho}^3_{2,k-1} \tag{A.12c}$$
$$\boldsymbol{\varphi}^4_k = \boldsymbol{\varphi}^3_{k-1} + d\boldsymbol{R}_3 \boldsymbol{\varrho}^3_{3,k-1} - d\boldsymbol{R}_3 d\boldsymbol{R}_4 \boldsymbol{\varrho}^4_{3,k-1}, \tag{A.12d}$$

and the corresponding updates to the redundant directors of each body, $\boldsymbol{d}^j_I$ are

$$\boldsymbol{d}_{I,k}^{1} = d\boldsymbol{R}_{1}\boldsymbol{d}_{I,k-1}^{1} \tag{A.13a}$$

$$\boldsymbol{d}_{I,k}^{2} = d\boldsymbol{R}_{1}d\boldsymbol{R}_{2}\boldsymbol{d}_{I,k-1}^{2} \tag{A.13b}$$

$$\boldsymbol{d}_{I,k}^{3} = d\boldsymbol{R}_{3}\boldsymbol{d}_{I,k-1}^{3} \tag{A.13c}$$

$$\boldsymbol{d}_{I,k}^{4} = d\boldsymbol{R}_{3}d\boldsymbol{R}_{4}\boldsymbol{d}_{I,k-1}^{4}. \tag{A.13d}$$

B AAReST Experimental Apparatus

This section summarizes the characterization to-date of the air table experimental apparatus and the prototype electromagnets for the docking system (see Figure 3.91).

B.0.4 Overview

As previously mentioned, the hardware and the majority of the software for the experimental apparatus was designed, built, and implemented by Dr. Marin Kobilarov, a KISS* post-doctoral fellow at Caltech. As shown in Figure 3.91, the apparatus consists of an airtable, two mobile robots, a desktop computer, and an overhead camera. The thrust-vectored fans were use to quantify the disturbances in the air table's pressure field.

B.0.5 Airtable

The airtable dimensions are 80cm×50cm in what will be called the $x-$ and $y-$directions, respectively. The manufacturer, Ealing Precision Student, specifies a flatness to within 0.005in, and the table can be leveled using four adjustable legs located in the corners. Compressed air is supplied at the at two inlets centered on the shorter side of the table, which can be independently regulated. Even with the table leveled and inlet pressure equal, the pressure field on the air table is not uniform. This can be seen simply from resting a puck on the table, and then starting the air supply–if there is no pressure gradient, the puck should not move. It is necessary to quantify the magnitude of the force produced by the uneven pressure distribution on the airtable in order to understand the testable range of the electromagnetic actuation system, and to determine whether sufficient attenuation measures can be taken to mitigate the effects of the disturbance field.

To this end, a direct quantification of the pressure field was undertaken with the use of the thrust vectored fans ans a simple proportional controller running at 20Hz, with feedback from the overhead

*KISS is the Keck Institute for Space Studies.

camera. Specifically, the CubeSat robot was commanded to go to a specific configuration in a grid with 5cm spacing, and stay there with zero velocity. Once the CubeSat stabilized to the specified state, the necessary control forces to maintain the state, i.e. the forces needed to opposed the pressure gradient, were recorded over a period of 5s. Algorithm 4 outlines this approach. Based on the close proximity (2.5cm) of the successive states, $\boldsymbol{s}_i$, and ample observation, it was determined that 35s was more than a sufficient amount of time from state $\boldsymbol{s}_i$ to state $\boldsymbol{s}_{i+1}$.

Algorithm 4 Airtable Force Field Determination

while $i <$ total states **do**
  Start timer $t = 0$s
  Go to state $\boldsymbol{s}_i$
  if $35s \leq t < 40$s **then**
    Output time t, control force $\boldsymbol{f}_c$, and state $\boldsymbol{s}_i$ to file
  end if
end while

The resulting mean force fields $x > -7.5$cm[†] are shown in Figure B.1, and are consistent with observed unactuated motions on the air table.

Disturbance Attenuation? An effort was made to used the recorded data in an open loop controller to mitigate the disturbance forces due to the air table. The algorithm was tested in an analogous way to the data collection algorithm. Specifically, the mobile robot was commanded using a proportional controller to go to a certain zero-velocity state. Once stabilized, the fans were shut off and the robot allowed to drift according to the natural dynamics of the airtable for 5s. Then, the robot was returned to the same state. Once *re*-stabilized, an open loop controller based on the recorded pressure force data was run for 5s, and the motion of the robot recorded.

Algorithm 5 Open Loop Disturbance Attenuation Test

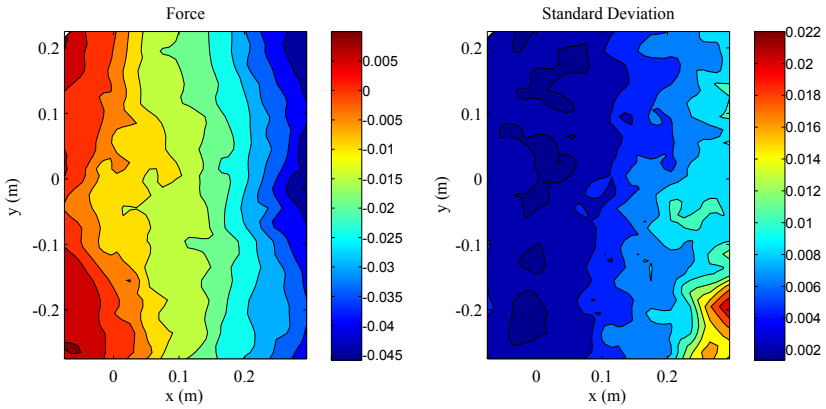
while $i <$ total states **do**
  Start timer $t_1 = 0$s
  Go to state $\boldsymbol{s}_i$
  if $35s \leq t_1 < 40$s **then**
    Output time t_1 and state $\boldsymbol{s}_i$ to file
  end if
  Start timer $t_2 = 0$s
  Go to state $\boldsymbol{s}_i$
  if $35s \leq t_2 < 40$s **then**
    Apply control force $\boldsymbol{f}_c$ from Algorithm 4
    Output time t_2 and state $\boldsymbol{s}_i$ to file
  end if
end while

The results of this test are shown in Figure B.2, and essentially reflect whether the use of the (either

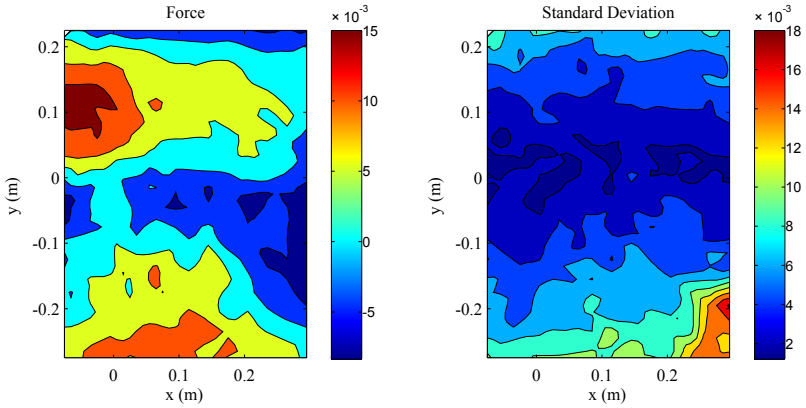
[†]The force field was only quantified in detail for this portion of the table because a frame to fix one of the robots to test the docking system precluded the other portion of the table from being use.

interpolated or fitted) mean recorded force is a close enough to the instantaneous force to effectively cancel out the disturbances by determining whether the drift of the robot is significantly reduced by the application of these forces. Unfortunately, as the variation in the initial data suggests (see Figure B.1), the answer is no, as will be discussed in the following paragraph.

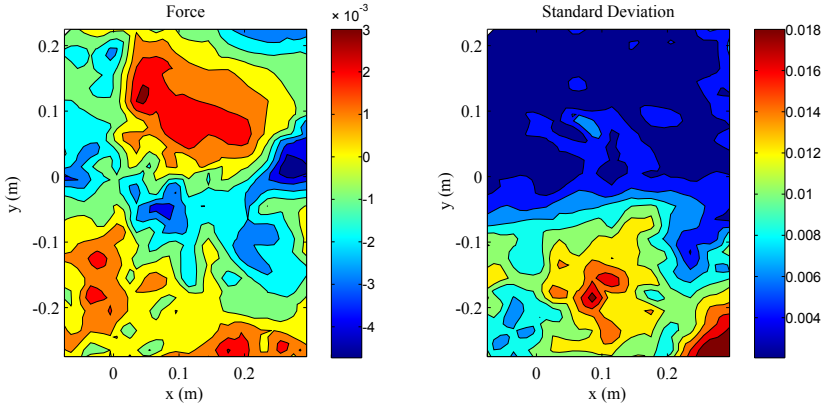
Comments. It is postulated that the variation in the air table pressure field is due to a combination of several factors, including, but not limited to, variation in the temperature of the compressed air and unsteady effects related to the internal baffling. This variation over time of the airtable forces is troublesome and renders and open loop disturbance attenuation strategy infeasible. However, the apparatus is still a useful tool to study dynamics in which the disturbances are dominated by other forces in the system.



(a) $x-$direction force field

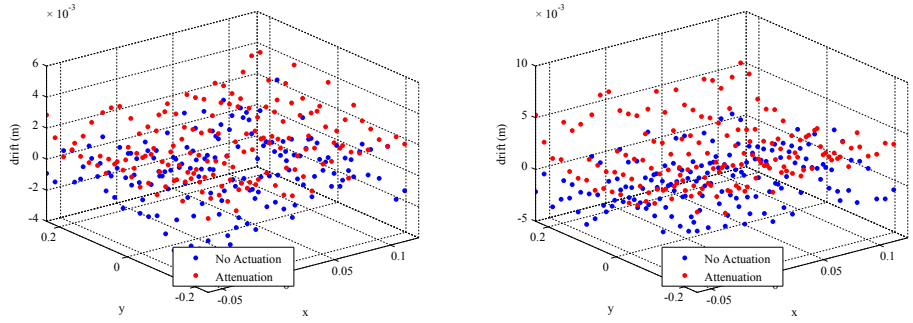


(b) $y-$direction force field



(c) $\theta-$direction force field

Figure B.1: Airtable disturbance field measurements according to Algorithm 4; units are (N) and (N-m) as appropriate.



(a) $x-$direction drift

(b) $y-$direction drift

Figure B.2: Recorded drift in $x-$ and $y-$directions with and without open loop disturbance attenuation. Ideally, the drift with the attenuation algorithm running should be significantly smaller in magnitude than the drift due to the natural dynamics of the air table.

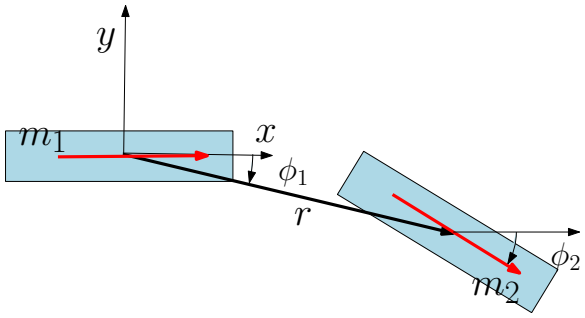


Figure B.3: Parameters used in magnetic model fit.

B.0.6 Electromagnets

The electromagnet prototypes were manufactured and supplied by the University of Surrey. The magnets consist of approximately 90m of 0.5mm diameter polyester coated copper wire, wrapped in approximated 9 layers at about 160 turns per coil, for a total thickness of 0.13cm. At a resistance of about 8 Ohms per coil, they draw about $600 - 650$ mA each. The 1cm diameter $\times$ 10cm core is Aperam Imphy Alloys' Supra 50, which is an Fe-Ni soft magnetic alloy. At a current of 600mA, the magnitude of the magnetic field in the code is 1T, and the permeability $\mu_s = 1$. Because of the length scales in the problem (10cm long solenoids interacting over distances of the same order or smaller), the point dipole model is not valid. To this end, a series of force calculations from first principles was made for a pair of interacting electromagnets in a range of likely relative orientations using a finite difference code developed in Matlab and contributed by Mathieu Blanchard and a function of the form

$$f_i = \frac{A_i(\phi_1, \phi_2)}{r^8} - \frac{B_i(\phi_1, \phi_2)}{r^4}. \tag{B.1}$$

was fit to the forces and torques on magnet m_1 due to magnet m_2, in the body-fixed frame of m_1, as shown in Figure B.3. In the final fit, $A(\phi_1, \phi_2)$ and $B(\phi_1, \phi_2)$ are fourth order polynomials. The direct calculations and fit were limited to the range $r \in [0.1, 0.55]$, $\phi_1 \in \left[\frac{-\pi}{5}, \frac{\pi}{5}\right]$ (making use of symmetry), and $\phi_2 \in \left[\frac{-\pi}{5}, \frac{\pi}{5}\right]$, based on the range in which the onboard visual posed estimation system is able to give state feedback for the anticipated controlled docking maneuver. Examples of the calculated forces, fitted forces, and the L_1-error between the two are shown in Figures B.4, B.5, and B.6.

The final values for the 4^{th} order polynomial coefficients for the two dimensional models are given in Table B.1. For use in space, the full 3d field needs to be quantified.

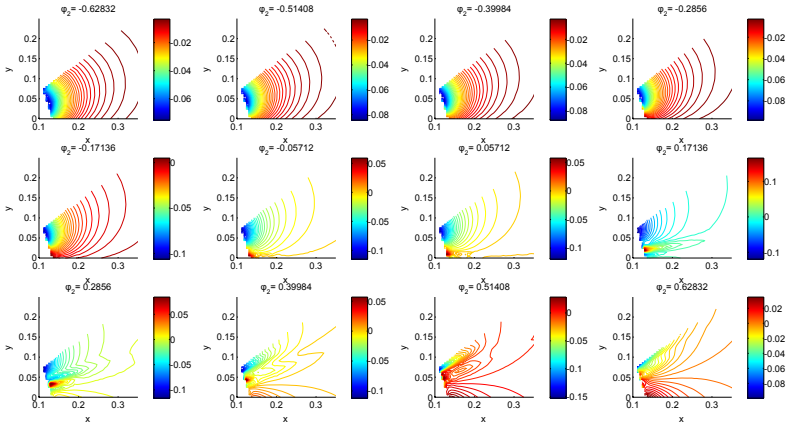


Figure B.4: Calculated magnetic forces in $y-$direction, N.

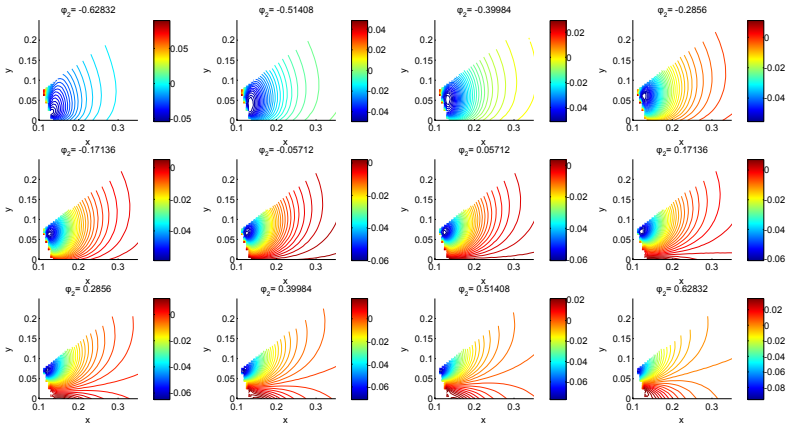


Figure B.5: Fit to calculated forces in $y-$direction, N.

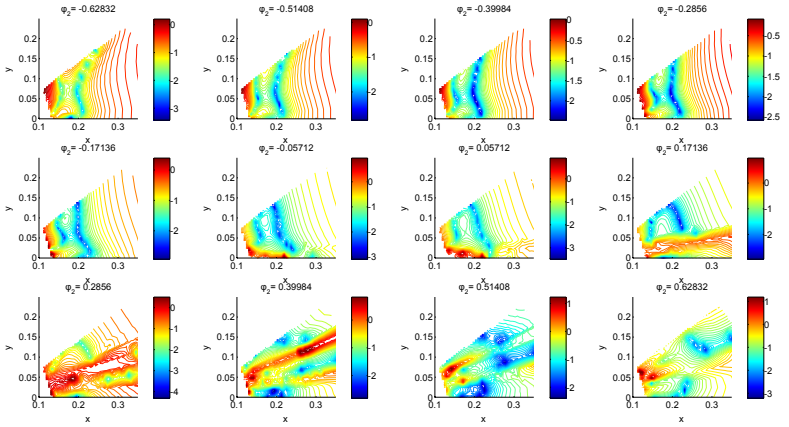


Figure B.6: L_1-error between calculation and fit, plotted in a log scale. This error metric is generally $< 10\%$ over the range of interest.

ϕ_1	ϕ_2	A_x	B_x	A_y	B_y	A_θ	B_θ
0	0	1.0670	-1.2094e-1	6.4039e-2	4.7942e-3	3.7875e-3	2.3510e-4
1	0	3.9386e-1	1.9773e-2	-1.3789	-4.8787e-1	-5.7298e-2	-1.6165e-3
0	1	-2.2751	-8.6915e-2	1.0495	1.2083	1.3477e-1	2.6519e-3
2	0	-3.5943	-1.2634e-2	-5.0010e-1	-5.8262e-2	-5.0081e-2	-3.3933e-3
1	1	9.5071	-1.8160e-1	6.9851	1.8216e-1	1.7989e-1	1.1954e-2
0	2	-7.8914	4.4923e-1	1.6582	-2.0492e-1	-2.5131e-1	-2.4009e-2
3	0	-1.5941	-7.5574e-2	-1.5941e1	2.5680e-1	9.5810e-2	3.9108e-3
2	1	1.5989e1	5.9169e-1	-3.0040	-1.3194	-3.2429e-1	-9.8745e-3
1	2	-3.0049e1	-8.7829e-1	-1.2417e1	2.4993	3.9376e-1	-2.1262e-2
0	3	2.0957e1	5.5594e-2	-6.0380e1	-7.2965e-1	-2.0751e-1	4.5830e-2
4	0	1.8429	4.7644e-2	8.2519e-1	8.8162e-2	9.7361e-2	6.5745e-3
3	1	1.5481	-1.0365	-6.8201	-7.9217e-1	-3.6091e-1	-1.6871e-2
2	2	-1.4833e1	-1.0365	5.4298	1.6634	7.8222e-1	2.8853e-2
1	3	2.1632e1	1.4074	5.2455	-2.2440	-7.6435e-1	9.3861e-3
0	4	-1.2404e1	-8.9385	6.0322e1	4.1154e-1	3.9942e-1	-2.2715e-2

Table B.1: Polynomial coefficients for 4^{th} order fit to the force field of two interacting solenoid magnets of the form in Equation B.1. The first two columns tabulate the exponent on ϕ_1 and ϕ_2, respectively.

Bibliography

[1] M. Akgul. A note on shadow prices in linear programming. *The Journal of the Operations Research Society*, 35(5):425–431, 1984.

[2] A. Akgunduz, P. Banerjee, and S. Mehrotra. A linear programming solution for exact collision detection. *Transactions of the ASME*, 5:48–55, 2005.

[3] M.D.S. Aliyu and K.S. Al-Sultan. Lp-based algorithm for detecting the collision of moving objects. *The Journal of the Operations Research Society*, 46(7):854–866, 1995.

[4] J. E. Andrade, K. W. Lin, C. F. Avial, and I. Vlahnich. Granular element for computational particle methods. *Computer Methods in Applied Mechanics and Engineering*, In Press, 2012.

[5] D.C. Aucamp and D.I. Steinberg. The computation of shadow prices in linear programming. *The Journal of the Operations Research Society*, 33(6):557–565, 1982.

[6] A. Bacciotti and L. Rosier. *Liapunov Functions and Stability in Control Theory.* Springer-Verlag: Berlin, Germany, 2005.

[7] O. Bachau and L. Trainelli. The vectorial paremetrization of rotation. *Nonlinear Dynamics*, 1: 71–92, 2003.

[8] M. Best. A compact formulation of an elastoplastic analysis problem. *Journal of Optimization Theory and Applications*, 37(3):343–353, 1982.

[9] P. Betsch. The discrete null space method for the energy consistent integration of constrained mechanical systems, part I: Holonomic constraints. *Computer Methods for Applied Mechanics and Engineering*, 194(50-52):5159–5190, 2005.

[10] P. Betsch and S. Leyendecker. The discrete null space method for the energy consistent integration of constrained mechanical systems, part II: Multibody dynamics. *International Journal for Numerical Methods in Engineering*, 67:499–552, 2006.

[11] P. Betsch and P. Steinmann. Constrained integration of rigid body dynamics. *Computer Methods in Applied Mechanics and Engineering*, 191:467–488, 2001.

[12] F. Bullo and A. D. Lewis. *Geometric Control oc Mechanical Systems.* Springer: New York, NY, 2005.

[13] N. Chakraborty, J. Peng, S. Akella, and J.E. Mitchell. Proximity queries between convex objects: An interior point approach for implicit surfaces. *IEEE Transactions in Robotics*, 24 (1):211–220, 2008.

[14] K. Chung and W. Wang. Quick collision detection of polytopes in virtual environments. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology*, pages 125–131, 1996.

[15] F. Cirak and M. West. Decomposition contact response (dcr) for explicit finite element dynamics. *International Jounrnal for Numerical Methods in Engineering*, 64, 2005.

[16] F.H. Clarke. *Optimization and Non-smooth Analysis.* Society for Industrial and Applied Mathematics, 1990.

[17] J. Cohen, M. Lin, D. Manocha, and K. Ponamgi. I-collide: An interactive and exact collision detection system for large-scaled environments. In *Proceedings of ACM Institute on 3D Graphics Conference*, pages 189–196, 1995.

[18] N. Correll and A. Martinoli. Modeling and designing self-organized aggregation in a swarm of miniature robots. *International Journal of Robotics Research*, 30(5):615–626, 2011.

[19] G.B. Dantzig and M.N. Thapa. *Linear Programming 1: Introduction.* Springer, 1997.

[20] D.P. Dobkin and D.G. Kirkpatrick. A linear algorithm for determining the separation of convex polyhedra. *Journal of Algorithms*, 6:381–392, 1985.

[21] H.A. Eishelt and C.-L. Sandblom. *Linear Programming and its Applications.* Springer, 2007.

[22] C. Ericson. *Real-Time Collision Detection.* Elsevier, 2005.

[23] J. Everist, K. Mogharei, H. Suri, and M. Ranasinghe. A system of in-space assembly. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2356–2361, 2004.

[24] E. O. Exler, T. Lehman, and K. Schittkowski. A comparative study of SQP-type algorthms for nonlinear and nonconvex mixed-integer optimization. *Submitted for publication*, 2011.

[25] R.C. Fetecau, J.E. Marsden, M. Ortiz, and M. West. Nonsmooth lagrangian mechanics and variational collision integrators. *Journal of Applied Dynamical Systems*, 2(3):381–416, 2003.

[26] R.M. Freund. The sensitivity of a linear program to changes in matrix coefficients. Working Paper 1532-84, MIT Sloan School of Business, 1984.

[27] R.M. Freund. Postoptimal analysis of a linear program under simultaneous changes in matrix coefficients. *Mathematical Programming Study 24*, pages 1–13, 1985.

[28] T. Gal. Shadow prices and sensitivity analysis in linear programming. *Operations Research Spectrum*, 8:59–71, 1986.

[29] E. G. Gilbert, D.W. Johnson, and S.S. Keerthi. A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Animation*, 43(2):193–203, 1988.

[30] E.G. Gilbert and S.M. Hong. A new algorithm for detection the collision of moving objects. *Robotics & Automation*, 1:8–14, 1989.

[31] H. Goldstein, C. Poole, and J. Safko. *Classical Mechanics*. Addison-Wesley: Reading, MA, 2002.

[32] M. Gonzalez, B. Schmidt, and M. Ortiz. Force-stepping integrators in lagrangian mechanics. *International Jounrnal for Numerical Methods in Engineering*, 84:1407–1450, 2010.

[33] S. Gottschalk, M.C. Lin, and D. Manocha. Obbtree: a hierarchical structure for rapid interference detection. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, pages 171–180, 1996.

[34] J.-B. Hiriart-Urruty and C. Lemarechal. *Fundamentals of Convex Analysis*. Springer, 2000.

[35] S. Hormoz and M. P. Brenner. Design principles for self-assmebly with short range interactions. *PNAS*, 108(13):5193–5198, 2011.

[36] P. Jimenez, F. Thomas, and C. Torras. 3d collision detection: a survey. *Computers & Graphics*, 25:289–285, 2001.

[37] I. Joyner, L. B. Holder, and D. J. Cook. Concept formation using graph grammars. In *Proceedings of the KDD Workshop on Multi-Relational Data Mining*, 2002.

[38] C. Kane, E.A. Repetto, M. Ortiz, and J.E. Marsden. Finite element analysis of non-smooth contact. *Computer Methods in Applied Mechanics and Engineering*, 180(1-2):1–26, 1999.

[39] C. Kane, J.E. Marsden, M. Ortiz, and M. West. Variational integrators and the newmark algorithm for conservative and dissipative mechanical systems. *International Journal for Numerical Methods in Engineering*, 49:1295–1325, 2000.

[40] S. Kenyon, C. Bridges, D. Liddle, B. Dyer, J. PArsons, M. Pollard, D. Feltham, R. Taylor, D. Mellor, A. Schofield, R. Linehan, R. Long, J. Fernandez, H. Kadhem, P. Davies, J. Gebbie, N. Holt, P. Shaw, L. Visagie, T. Theodorou, V. Lappas, and C. Underwood. STRAND-1: Use of a $500 smartphone as the central avionics of a nanosatellite. In *Proceedings of the 62nd International Astronautical Congress*, 2011.

[41] C. Klapproth, P. Deuflhard, and A. Scheila. A perturbation result for dynamical contact problems. Technical Report 08-27, Konrad-Zuse-Zentrum fur Informationstechnik Berlin, 2008.

[42] E. Klavins. Programmable self-assmbly. *IEEE Control Systms Magazine*, August:43–56, 2007.

[43] Marin Kobilarov. Cross-entropy randomized motion planning. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

[44] P. Krysl. Dynamicslly equivalend implicit algorithms for the integration of rigid body rotations. *Communications in Numerical Methods in Engineering*, 24(2):141–156, 2008.

[45] P. Krysl and L. Endres. Explicit Newmark/Verlet algorithm for time integration of the rotational dynamics if rigid bodies. *International Journal for Numerical Methods in Engineering*, 62:2154–2177, 2005.

[46] R. I. Leine, C. Glocker, and D .H. van Campen. Nonlinear dynamics and modeling of various wooden toys with impact and friction. *Journal of Vibration and Control*, 9:35–78, 2003.

[47] A. Lew, J. Marsden, M. Ortiz, and M. West. Asynchronous variational integrators. *Archives for Rational Mehanics and Analysis*, 167:85–146, 2003.

[48] S. Leyendecker. *Mechanical integrators for constrained dynamics systems in flexible multibody dynamics*. PhD thesis, University of Kaiserslautern, 2006.

[49] S. Leyendecker, P. Betsch, and P. Steinmann. Objective energy-momentum conserving integration for the constrained dynamics of geometrically exact beams. *Computer Methods in Applied Mechanics and Engineering*, 195:2313–2333, 2006.

[50] S. Leyendecker, J.E. Marsden, and M. Ortiz. Variational integrators for constrained dynamical systems. *ZAMM*, 88(9):677–708, 2008.

[51] S. Leyendecker, L.J. Lucas, H.Owhadi, and M. Ortiz. Optimal constrol strategies for rubust certification. *ASME Journal of Computational and Nonlinear Dynamics*, 5(6):505–528, 2010.

[52] S. Leyendecker, S.Ober-Blobaum, J.E. Marsden, and M. Ortiz. Discrete mechanics and optimal control for constrained systems. *Optimal Control Applications and Methods*, 31, 2010.

[53] S. Leyendecker, C. Hartmann, and M. Koch. Variational collision integrator for polymer chains. *Journal of Computational Physics*, 231:3896–3911, 2012.

[54] C.F. Li, Y.T. Feng, and D.R.J Owen. Smb: Collision detection based on temporal coherence. *Computer Methods in Applied Mechanics and Engineering*, 195:2252–2269, 2006.

[55] C.-J. Lin. Computing shadow prices/costs of degenerate lp problems with reduced simplex tables. *Expert Systems with Applications*, 37:5848–5855, 2010.

[56] M.C. Lin, D. Manocha, and J. Canny. Fast contact determination in dynamic environments. In *Proceedings of the IEEE International Conference on Robotics and Animation*, pages 602–608, 1994.

[57] J.E. Marsden and T.S. Ratiu. *Introduction to Mechanics and Symmetry.* Springer, second edition, 1999.

[58] J.E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, 10:223–246, 2001.

[59] H.D. Mills. Marginal values of matrix games and linear programs. In H.W. Kuhn and A.W. Tucker, editors, *Linear Inequalities and Related Systems*, pages 183–193. Princeton University Press, 1956.

[60] A. Moore. *Discrete Mechanics and Optimal Control for Space Trajectory Design.* PhD thesis, California Institute of Technology, 2011.

[61] A. Moore, S. Ober-Blöbaum, and J. E. Marsden. Trajectory design combinding invariant manifolds with discrete mechanics and optimal control. *To appear in: Journal of Guidance and Control*, 2012.

[62] S. Murata, H. Kurokawa, K. Tomita, and S. Kokaji. Modeling and designing self-organized aggregation in a swarm of miniature robots. *Artificial Life Robotics*, 1:111–115, 1997.

[63] S. Murata, H. Kurokawa, E. Yoshida, S. Kokaji, and K. Tomita. A 3-d self-reconfigurable structure. In *IEEE International Conference on Robotics and Automation*, pages 432–439, 1998.

[64] N. Napp and E. Klavins. A compositional frameworkd for programming stochastically interacting robotics. *International Journal of Robotics Research*, 30(6):713–729, 2011.

[65] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, 1999.

[66] S. Ober-Blöbaum, O. Junge, and J. Marsden. Discrete mechanics and optimal control: an analysis. *ESAIM: Control, Optimisation and Calculus of Variations*, 17:322–352, 2011.

[67] H. Owhadi, T. J. Sullivan, M. McKerns, and M. Ortiz and. Optimal uncertainty quantification. *SIAM Review*, In press, 2012.

[68] A. Pandolfi, C. Kane, J.E. Marsden, and M. Ortiz. Time-discretized variational formulation of non-smooth frictional contact. *International Jounrnal for Numerical Methods in Engineering*, 53(8):1801–1829, 2002.

[69] S. Pankavich, Z. Shreif, Y. Miao, and P. Ortoleva. Self-assembly of nanocomponents into composite structures: Derivation and simulation of the langevin equations. *The Journal of Chemical Physics*, 130(194115):1–10, 2009.

[70] K. Patterson, N. Yamamoto, and S. Pellegrino. Thin deformable mirrors for a reconfigurable space telescope. In *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2012.

[71] J. Peet, A. J. Heeger, and G. C. Bazan. 'Plastic' solar cells: Self-assembly of bulk heterojunction nanomaterials by spontaneous phase separation. *Accounts of Chemical Research*, 42(1):1700–1708, 2009.

[72] R. A. Peters, K. A. Hambuchen, and R. E. Bodenheimer. The sensory ego-sphere: A mediating interface between sensors and cognition. *Autonomous Robots*, 26(1):1–19, 2009.

[73] F. Pfeiffer and C. Glocker. *Multibody dynamics with unilateral contacts*. John Wiley & Sons: New York, NY, 1996.

[74] F. Radjai and V. Richefeu. Contact dynamics as a nonsmooth discrete element method. *Mechanics of Materials*, 41:715–728, 2009.

[75] R.T. Rockafellar. *Convex Analysis*. Princeton University Press, 1970.

[76] M. Schechter. Unrestricted variables in linear programming. *Journal of Optimization Theory and Applications*, 69(3):605–610, 1991.

[77] R. Seidel. Linear programming and convex hulls made easy. In *Proceedings of the Sixth Annual ACM Conference on Computational Geometry*, pages 211–215, 1990.

[78] J. P. Shoer and M. A. Peck. A flux-pinned magnet-superconductor pair for close-proximity station keeping and self-assmbly of spacecraft. In *Proceedings of the 2007 AIAA Guidance, Naviation, and Control Conference and Exhibit*, 2007.

[79] M. J. Solomon, R. Zeitoun, D. Ortiz, K. E. Sung, D. Deng, A. Shah, M. A. Burns, S. C. Glotzer, and J. M. Millunchick. Toward assembly of non-close-packed colliodal structures from anisotropic pentamer particles. *Macromolecular Rapid Communications*, 31(2):196–201, 2010.

[80] R. A. Spurrier. Comments on 'singularity-free extraction of a quaternion from a direction-cosine matrix'. *Journal of Spacecraft and Robotics*, 13:754–755, 1978.

[81] M. Tang, S. Curtis, S.-E. Yoon, and D. Manocha. Iccd: Interactive continuous collision detection between deformable models using connectivity-based culling. *IEEE Transactions on Visualizationa and Computer Graphics*, 15(4):544–557, 2009.

[82] M. T. Tolley and H. Lipson. On-line assembly planning for stochastically reconfigurable systems. *International Journal of Robotics Research*, 30(13):1566–1584, 2011.

[83] P. Tsiortras. Stabilization and optimality results for the attitude control problem. *Journal of Guidance, Control, and Dynamics*, 19(4):1–9, 1996.

[84] G. M. Whitesides and B. Gryzbowski. Self assembly at all scales. *Science*, 295:2418–2412, 2002.

[85] C. Wieland. *Computational Space Flight Mechanics.* Springer-Verlag: Heidelberg, Germany, 2010.

[86] W. E. Wiesel. *Spaceflight Dynamics.* Aphelion Press: Beavercreak, OH, 3rd edition, 2010.

[87] A.C. Williams. Marginal values in linear programming. *Journal of the Society for Industrial and Applied Mathematics*, 11(1):82–94, 1983.

[88] K. Woellert, P. Ehrenfreund, A. J. Ricco, and H. Hertzfeld. CubeSats: Cost-effective science and technology platforms for emerging and developing nations. *Advances in Space Research*, 47:663–684, 2011.