Approximating Contact for Multi-End Effector Grasps

by

Abdesselam GUERROUDJ

THESIS PRESENTED TO ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
IN PARTIAL FULFILLMENT OF A MASTER'S DEGREE
WITH THESIS IN SOFTWARE ENGINEERING
M.A.Sc.

MONTREAL, AUGUST 25, 2022

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

BOARD OF EXAMINERS

THIS THESIS HAS BEEN EVALUATED

BY THE FOLLOWING BOARD OF EXAMINERS

Dr. Sheldon Andrews, Thesis supervisor
Software and Information Technology Engineering Department, École de technologie supérieure

Dr. David Labbé, Chair, Board of Examiners
Software and Information Technology Engineering Department, École de technologie supérieure

Dr. Paul G. Kry, External Examiner
School of Computer Science, McGill University

THIS THESIS WAS PRESENTED AND DEFENDED

IN THE PRESENCE OF A BOARD OF EXAMINERS AND THE PUBLIC

ON JULY 29, 2022

AT ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

FOREWORD

This thesis was completed as part of my Master's degree in Software Engineering at the École de technologie supérieure (ÉTS) in Montreal, Canada. This thesis is on multi-finger grasping approximation. I'd like to thank a few people here after thanking my family and close friends for their unending support. Dr. Sheldon Andrews, my supervisor, for recommending this research topic and for his help throughout the road, and Dr. David Labbé, for his advice and recommendations. Working with these folks was a steep learning curve for me because they not only polished my research abilities but also provided me with an understanding of the dimensions of the animation and simulation sectors. I'd also like to thank the ÉTS staff for their assistance and support throughout the journey.

Approximation du contact pour la préhension des effecteurs multiples

Abdesselam GUERROUDJ

RÉSUMÉ

Dans cette thèse, une nouvelle approche pour simuler la préhension de multi-effecteurs en temps réel par approximation est présenté. En considérant un objet entièrement saisi et immobilisé, il est démontré comment multiples contraintes de contact et de frottement peuvent être remplacées par une seul contrainte de 6DOF, ce qui permet d'approximer le comportement de la préhension et d'améliorer considérablement les performances. De plus, il est démontré à quel point le comportement de préhension approximatif est similaire au comportement entièrement simulé tout en offrant une augmentation considérable des performances. Ceci est démontré en testant notre approche dans deux scénarios distincts utilisant un bras et une main simulés saisissant deux objets différents. De plus, il est démontré comment des axes spécifiques de la contrainte 6DOF peuvent être relâchés pour imiter les mouvements de friction et de glissement de l'objet cible par rapport aux effecteurs.

Mots-clés: Simulation, simuluation de la préhension, préhension multi-effecteurs

Approximating Contact for Multi-End Effector Grasps

Abdesselam GUERROUDJ

ABSTRACT

We present a novel approach for simulating a multi-end effector grasp in real-time by approximation. Given a fully grasped and constrained object, we show how multiple contact and friction constraints can be replaced by a single 6DOF that approximates the grasping behavior and significantly improves performance.

We demonstrate how similar the approximated grasp behavior is to the fully simulated behavior while providing a considerable performance increase by testing our approach in two separate scenarios using a simulated arm and hand grasping two different objects. Furthermore, we show how specific axes of the 6DOF constraint can be relaxed to imitate friction and sliding motions of the target object relative to the end-effectors.

Keywords: Physics simulation, grasp simulation, grasp approximation, multi end-effector grasp

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OF ABBREVIATIONS

MBS	Multi-body system
RBD	Rigid-body dynamics
DOF	Degrees of freedom

CHAPTER 1

INTRODUCTION

Interaction behavior synthesis has been one main focus in robotics and 3D characters research. In interactive and real-time computer graphics applications, such as video games, complex environments make encoding all character-environmental interactions particularly challenging and time-consuming. For instance, it would be laborious to record every possible character-environment interaction via motion capture. Real-time physics simulation for different interactive actions such as grabbing and manipulation objects is the solution. For real-time applications, however, it is necessary to use physics simulation cautiously. The performance may suffer if complicated interactive activities need a sizable number of computations to be run at each simulation time step.

In this thesis, we are focused on grasping, a crucial human action that is challenging and complex to simulate. A grasp occurs when a hand makes contact with an object at multiple points with the goal to transport or manipulate an object.

1.1 Motivation

Interactions between humanoid characters and virtual worlds present two challenges. A grasp must appear physically plausible and lifelike, which can be obtained by real-time simulation. However, the performance impact of simulating a grasp must be minimal. From a computational standpoint, grasping simulation is challenging. This is because it involves ill-conditioned and degenerate linear systems. Furthermore, in a multi-body simulation for each new contact a grasp forms, a contact constraint that require solving is added to the simulation. Therefore, the greater the number of contacts involved in a grasp, the greater the detrimental influence on simulation performance. This is why simulation of contact-rich phenomenon real-time such as grasping, is a challenge even when using expensive state-of-the-art hardware.

Consequently, the purpose of this thesis is to address the issue of the excessive contacts that are intrinsically linked to grasping and the detrimental effects they have on performance.

1.2 Research Problem

In this thesis, our primary motivation is to accelerate contact-rich grasp simulation in real-time applications. While preserving the visual quality of the ground truth simulation.

1.3 Contributions

We present a method that approximates a grasp. Which behaves visually similar to the fully simulated grasp, while reducing the computations of all contact constraints involved. We achieve this by replacing all contact constraints with a single 6DOF constraint. We then relax specific axes of the constraint to imitate friction and relative sliding motion that may present when a grasp encounters external perturbations.

1.4 Thesis Organization

The remainder of this thesis is structured as follows:

- **Chapter 1 and 2** present the theoretical background for our methodology and the literature review of related work in the areas of grasp simulation and approximation;
- **Chapter 3** presents our contribution, a simple and efficient approach for approximating a contact-rich grasp;
- **Chapter 4** presents two experiments on our approach to demonstrate performance improvements and visual fidelity;
- **Conclusion** summarizes contributions and and discusses current limits of the method as well as potential future research avenues for improvement.

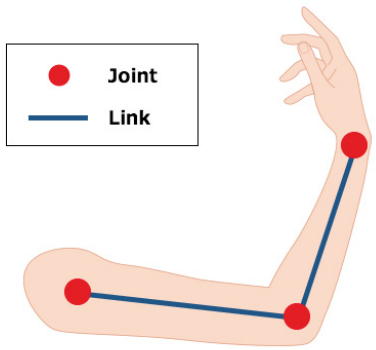
CHAPTER 2

BACKGROUND

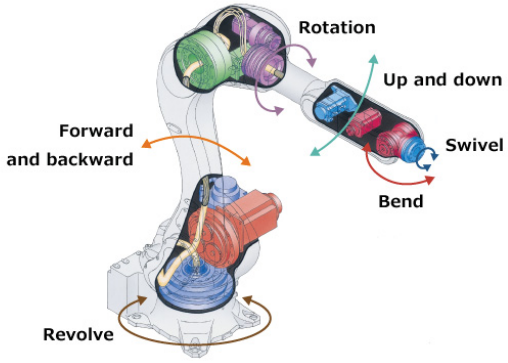
2.1 Multi-body Simulation

The main goal of physics simulation is to create virtual/digital objects that behave like their real-world counterparts. Objects can be rigid, such as a bowling ball or soft such as cloth. For the sake of simplicity in this thesis, rigid bodies are used. When two bodies are in contact, only one contact constraint forms when employing rigid bodies, while several redundant contact constraints can form when utilising soft bodies.

Bodies, joints, and force elements compose a multi-body system. These components can be used to derive the kinematics and dynamics of the entire system. Simple intuitive examples of multi-body systems (MBS) are the human arm and the robotic arm as shown in Figure 2.1.



a) Human arm example



b) MBS kinematic constraints example. [https://robotics.kawasaki.com]

Figure 2.1 Examples of Multi-body systems

A simulation calculates the forces, moments, and torques acting on each rigid body by accounting for both the impact of other parts of the system and external factors (gravity). The following section is a summary of the different formulations of frictional contact and methods to solve these problems (see also Andrews & Erleben's (2021)).

2.1.1 Equations of Motion

Newton's second law of motion expresses the net force $f_b(t)$ of acting on a body b at time t as a function of the body's mass m_b and its acceleration $\ddot{\mathbf{x}}_b$ written as follows

$$f_b(t) = m_b \ddot{\mathbf{x}}_b(t). \tag{2.1}$$

The dynamics of a multi-body system is generalized from Equation 2.1 similarly, by the Newton-Euler equations of motion [Goldstein, Poole & Safko 2002] expressed as

$$\mathbf{M}\ddot{\mathbf{x}}(t) = f(\mathbf{x}(t), \dot{\mathbf{x}}(t), t), \tag{2.2}$$

where $\mathbf{M}$ is a global matrix of body masses that is presumed to be constant in the local frame, $\mathbf{x}(t)$ is a vector of all positions and orientations of bodies, $\dot{\mathbf{x}}(t)$ is a vector of all velocities of bodies, all at time t. Equation 2.2 is a second-order ordinary differential equation. To use this equation in a computer simulation it needs to be solved numerically using a discrete numerical integration with a time step h. We outline the process below.

2.1.2 Discrete Time Integration

The velocities $\dot{\mathbf{x}}(t)$ can be expanded using Taylor series, we take the first order approximation

$$\dot{\mathbf{x}}_+ \approx \dot{\mathbf{x}} + h\ddot{\mathbf{x}} \tag{2.3}$$

where $\dot{\mathbf{x}}_+$ are the velocities at the next time step. This means that the velocities at the next time step are determined by the velocities and accelerations at the current time step.

Replacing $\ddot{\mathbf{x}}$ in Equation 2.2 and dropping time notation gives:

$$\mathbf{M}\dot{\mathbf{x}}_+ = \mathbf{M}\dot{\mathbf{x}} + h\mathbf{f} \tag{2.4}$$

In a multibody system with k bodies, and n degrees of freedom, the motion of this system can be expressed by Equation 2.4 where $\mathbf{M} \in \mathbb{R}^{n \times n}$ is a square $n \times n$ matrix of mass, $\mathbf{M}\dot{\mathbf{x}} \in \mathbb{R}^n$ is the momentum term, and $\mathbf{f} \in \mathbb{R}^n$ is the forces term. Notably, $n \leq 6k$.

2.1.3 Constraints

A body in motion that can translate and rotate freely in 3D space is said to have 6 degrees of freedom (DOF). Three positional DOF along x, y, and z axes and three rotational DOF about x, y, and z. A constraint can limit the body's motion (position or rotation) on one or more axes, lowering its DOF.

In a simulation, constraints are modeled as a collection of m constraint functions $g(\mathbf{x}) \in \mathbb{R}^m$. Constraint functions $g(\mathbf{x})$ can be of two forms:

1. **Bilateral constraint function** have the form $g(\mathbf{x}) = 0$. Ball-and-socket joints, and hinges are joints that are modeled using this form.
2. **Unilateral constraint function** has the form $g(\mathbf{x}) \geq 0$. Contact constraints are modeled using this form.

The motion of bodies can be coupled via constraints. A revolute (hinge) joint can be used to constrain the motion of two moving bodies so that only rotation around an anchor point is permitted. Finding the appropriate velocities that satisfy the constraints is necessary to simulate their coupled motion; otherwise, the integrator would permit the bodies to continue moving independently. To solve for velocities, we need to express $g(\mathbf{x})$ in terms of velocities by computing the gradient of $g(\mathbf{x})$ with respect to $\mathbf{x}$, as follows:

$$\mathbf{J} = \frac{\partial g(\mathbf{x})}{\partial \mathbf{x}} \in \mathbb{R}^{m \times n} \tag{2.5}$$

The term $\mathbf{J}$ is commonly referred to as the Jacobian matrix. The resulting velocity level constraint equations can be written as $\mathbf{J}\mathbf{x} = 0$, and $\mathbf{J}\dot{\mathbf{x}} \geq 0$ for unilateral and bilateral constraints respectively. The constraints are imposed on the system during simulation by including them as constraint

forces in the equation of motion. The constraint forces are computed as follows:

$$\mathbf{f}_c = \mathbf{J}^T \boldsymbol{\lambda} \tag{2.6}$$

The term $\boldsymbol{\lambda} \in \mathbb{R}^m$ is the magnitudes of the constraint forces, when the gradients are normalized. The terms $\boldsymbol{\lambda}$ are known as the Lagrange multipliers. We apply $h\mathbf{f}_c$, which is an impulse, and denote the corresponding impulse magnitude by $\boldsymbol{\lambda}_+ \in \mathbb{R}^m$. We can update Equation 2.4 to include the constraint impulses, such that

$$\mathbf{M}\dot{\mathbf{x}}_+ - \mathbf{J}^T\boldsymbol{\lambda}_+ = \mathbf{M}\dot{\mathbf{x}} + h\mathbf{f}. \tag{2.7}$$

At this point we can formulate a linear system that combines both the velocity constraint equations and equations of motion as

$$\begin{bmatrix} \mathbf{M} & -\mathbf{J}^T \\ \mathbf{J} & 0 \end{bmatrix} \begin{bmatrix} \dot{\mathbf{x}}_+ \\ \boldsymbol{\lambda}_+ \end{bmatrix} = \begin{bmatrix} \mathbf{M}\dot{\mathbf{x}} + h\mathbf{f} \\ 0 \end{bmatrix}. \tag{2.8}$$

Using the first row to solve for $\dot{\mathbf{x}}_+$ will result in the following reduced linear system of the form $\mathbf{A}\boldsymbol{\lambda}_+ + \mathbf{b} = 0$ such that $\mathbf{A} = [\mathbf{J}\mathbf{M}^{-1}\mathbf{J}^T]$ and $\mathbf{b} = \mathbf{J}\mathbf{M}^{-1}(\mathbf{M}\dot{\mathbf{x}} + hf)$. During simulation, the solver iterates to find $\boldsymbol{\lambda}_+$ and finally the full motion of the degrees of freedom of the multi-body system is computed during simulation using Equation 2.7.

2.1.4 Contact Constraints

Contact constraints are a special case of constraints, they are only active when two or more bodies come in contact with each other. In the case of rigid bodies, one contact constraint is formed between each two bodies in contact. A contact constraint is modeled by the constraint function g, also named the gap function. It measures the distance between the two bodies. If $g > 0$, the bodies are not in contact yet. If $g = 0$ the two bodies are in contact. The case of $g < 0$

indicates there is interpenetration between the two bodies, which should be impossible in the case of rigid bodies.

At the point of contact, we define a unit vector $\hat{n}$, facing away from one object, and orthogonal to the contact's tangent plane. To satisfy the contact constraint, an impulse of magnitude λ_n is applied at the contact point, and in the direction of $\hat{n}$. Suppose we are given two rigid bodies A and B at contact with each other at point c. The objects' centers of mass are located at $\mathbf{s}_A$ and $\mathbf{s}_B$ respectively. Bodies A and B have linear velocities $\dot{\mathbf{v}}_A$ and $\dot{\mathbf{v}}_B$ and angular velocities ω_A and ω_B. The vector $\hat{n}$ is the unit contact normal directed from A to B. The relative velocity $\dot{\mathbf{x}}_{\hat{n}}$ in direction of $\hat{n}$ is given by

$$\dot{\mathbf{x}}_{\hat{n}} = \hat{n} \cdot [(\dot{\mathbf{x}}_B + \omega_B \times (\boldsymbol{\lambda} - \mathbf{s}_B)) - (\dot{\mathbf{x}}_A + \omega_A \times (\mathbf{x} - \mathbf{s}_A)] \tag{2.9}$$

Setting $\mathbf{r}_A = (c - \mathbf{s}_A)$ and $\mathbf{r}_B = (c - \mathbf{s}_B)$ the skew-symmetric mass of a vector's cross product is defined as

$$r^{\times} = \begin{bmatrix} 0 & -z & y \\ z & 0 & -x \\ -y & x & 0 \end{bmatrix} \tag{2.10}$$

$$\dot{\mathbf{x}}_{\hat{n}} = \underbrace{\begin{bmatrix} -\hat{n}^T & \hat{n}^T \mathbf{r}_A^{\times} & \hat{n}^T & -\hat{n}^T \mathbf{r}_B^{\times} \end{bmatrix}}_{\mathrm{J}} \underbrace{\begin{bmatrix} \dot{\mathbf{x}}_A \\ \omega_A \\ \dot{\mathbf{x}}_B \\ \omega_B \end{bmatrix}}_{\dot{\mathbf{x}}}. \tag{2.11}$$

From Equations 2.8 and 2.11 we observe that the number of computations needed to solve the linear system is dependent on the size of the Jacobian matrix, which in turn is dependent on the number of constraints. This is the main reason we propose to reduce the number of constraints in this thesis, which reduces the number of computations necessary to solve for the constraints at each time step.

2.1.5 Friction

Consider the two bodies A and B mentioned in the previous section. When they are in contact, and not sliding, Coulomb's law of friction states there exists a friction impulse $\boldsymbol{\lambda}_t$ between the two objects expressed as a function of the normal impulse $\boldsymbol{\lambda}_{\hat{n}}$ such that

$$\|\boldsymbol{\lambda}_t\| \leq \mu \boldsymbol{\lambda}_n, \tag{2.12}$$

where μ is the coefficient of friction and is positive. The set of possible friction forces that satisfy the above inequality give a quadratic friction cone FC. Given a frame at the contact point, where the contact normal $\hat{n}$ is along the z axis, and x, and y, axes form the contact tangential plane, the set of friction forces is given by

$$FC = \left\{ \mathbf{f} \mid \mathbf{f}_z \geq 0,\ \sqrt{\mathbf{f}_x^2 + \mathbf{f}_y^2} \leq \mu \mathbf{f}_z \right\}. \tag{2.13}$$

It is common to approximate the friction cone using a linear approximation, as a polyhedral or box cone (see Figure 2.2). We use the Vortex SDK, and the friction cone is approximated as a box.

Each contact force creates a moment about coordinate frames that are different from the contact point's frame. These moments can be represented as a wrench cone corresponding to the friction cones for each contact such that

$$\mathbf{w} = \begin{bmatrix} c \times \mathbf{f} \\ \mathbf{f} \end{bmatrix}, \tag{2.14}$$

where c is the contact point vector . The set of all wrenches is a the positive span of $\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, \mathbf{w}_4$ the correspond to the friction cone edges forces (basis) $\mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \mathbf{f}_4$.

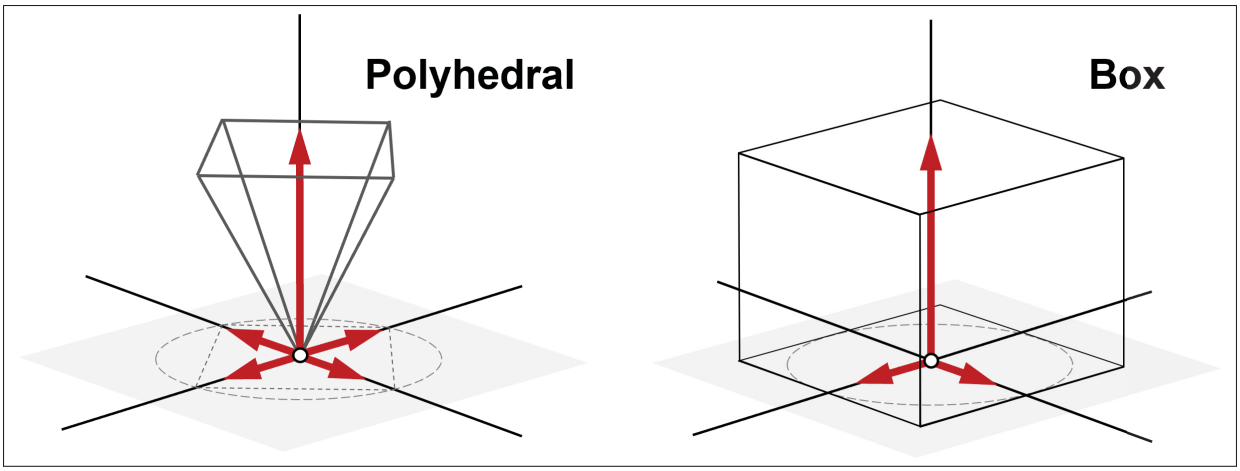


Figure 2.2 Linear models of friction: polyhedral cone (left) and box (right). The vertical axis scales with the non-interpenetration force in the normal direction, and the surface defined by a horizontal slice through each cone approximation gives the limits of the frictional forces. Modified from [Andrews & Erleben (2021)]

2.2 Grasping

In robotics and simulation, grasping and manipulating objects is an essential and challenging task to tackle, as well as an active research topic. A successful grasp aims to keep an object in the hand despite external disturbance without destroying it.

2.2.1 Grasp Modeling

Given a target object with its center of mass as frame o grasped at N contact points, according to Howard & Kumar's (1996) work each contact is classified as either frictionless, frictional, or soft. At each frictional contact i's location, a normal force and a tangential force are exerted on the object.

These contact wrenches are denoted by ${}^{b}_{i}\mathbf{w}_{n}$ and ${}^{b}_{i}\mathbf{w}_{t}$ respectively represented in contact point frame b corresponding to ith contact. The corresponding wrench intensities are given by ${}^{b}_{i}\boldsymbol{\lambda}_{n}$ and ${}^{b}_{i}\boldsymbol{\lambda}_{t}$. Similarly, a frictionless contact has a single wrench ${}^{b}_{i}\mathbf{w}_{n}$ and wrench intensity ${}^{b}_{i}\boldsymbol{\lambda}_{n}$. Each wrench can be expressed as

$$ {}^{b}_{i}\mathbf{w} = \begin{bmatrix} {}^{b}_{i}c \times {}^{b}_{i}\mathbf{f} \\ {}^{b}_{i}\mathbf{f} \end{bmatrix}, \tag{2.15} $$

such that ${}^{b}_{i}\mathbf{w}$ is a wrench that corresponds to a contact force ${}^{b}_{i}\mathbf{f}$ at contact point i with position ${}^{b}_{i}c$ in frame b.

In Howard & Kumar's (1996), the authors define a grasped object with frictional contacts and an external wrench $\mathbf{W}_g$ to be in equilibrium iff

$$ \begin{aligned} &1)\ \forall_i\ {}^{i}\boldsymbol{\lambda}_n \geq 0,\ |{}^{i}\boldsymbol{\lambda}_t| \leq {}^{i}\mu_t\ {}^{i}\boldsymbol{\lambda}_n \\ &2)\ \mathbf{W}\boldsymbol{\lambda} + \mathbf{W}_g = 0 \text{ for } \boldsymbol{\lambda} \neq 0\,, \end{aligned} \tag{2.16} $$

where ${}^{b}_{i}\mu$ is the coefficient of the tangential friction given by Coulomb's law, $\mathbf{W}$ is the wrench matrix composed of contact wrenches as columns, and $\boldsymbol{\lambda}$ is a vector of wrench intensities. Wrench intensities $\boldsymbol{\lambda}$ are internal forces that do not contribute to the acceleration of the grasped object. Grasp quality is a measure we use to decide when we disable contact constraints in a grasp. We estimate the grasp quality as follows, from Andrews & Kry's (2013) work. Firstly we map linear contact forces from world frame to wrenches in the object's frame. This is achieved by the matrix

$$ \mathbf{D}_i = [\mathbf{R}^T - \hat{c}_i\mathbf{R}^T]^T, \tag{2.17} $$

where $\mathbf{R}$ is the rotation matrix transforming vectors from world frame to object's frame and c_i is the contact location used to form the skew symmetric cross product matrix $\hat{c}_i$. The set of wrench vectors is then computed as

$$ \mathbf{w}_i = \begin{bmatrix} \mathbf{D}_i\,\mathbf{f}_{i_1} & \mathbf{D}_i\,\mathbf{f}_{i_2} & \mathbf{D}_i\,\mathbf{f}_{i_3} & \mathbf{D}_i\,\mathbf{f}_{i_4} \end{bmatrix}, \tag{2.18} $$

where $\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \mathbf{f}_{i_3}, \mathbf{f}_{i_4}$ are the basis forces for the i^{th} contact. It is worth mentioning that since we're using a symmetric box model of friction, some of those vectors are symmetric, e.g., $\mathbf{f}_{i_1} = -\mathbf{f}_{i_2}$ since the two vectors span the positive and negative directions. We then form the

matrix $\mathbf{G} = [\mathbf{w}_i \dots \mathbf{w}_N]$. Grasp quality is estimated as the smallest singular value of the singular value decomposition of $\mathbf{G}$.

If a rigid body has a set of frictional contacts that can generate a wrench $\mathbf{w}_f$ that can negate an external wrench w_g acting on the rigid body, the grasp is known as a force-closure grasp.

To mimic dexterity, frictional behavior must be visual. In other words, the object should be able to slide and rotate relative to end-effectors. Furthermore, the end-effectors must flex when moving or pulling the object.

When grasping with contact constraints enabled, the object's sliding behavior is proportional to the sum of contact forces applied on the object's surface. We provide further details about how friction is approximated using our approach in Chapter 3.

CHAPTER 3

LITERATURE REVIEW

Synthesizing human motion for virtual characters and robotics is an intriguing challenge. Gripping and dexterous manipulation are particularly interesting due to the increased articulation and multi-contact interactions.

In this thesis, we address the problem of simulating static grasps of an object securely gripped by a hand. A grasp simulation can pertain to a variety of factors given a set of end effectors, such as virtual hands and an object being grasped. Poses, forces, and contact points of end effectors relative to the grasping task, or object can all be used to construct a grab configuration. A number of researchers have proposed solutions to the problem of grasp planning and synthesis. This chapter covers not only work on grasping simulation and approximation, but also grasping synthesis and planning. The intent is to give a broader context of the work.

3.1 Grasping simulation

Many modern approaches simulate a grasp by coupling an articulated skeleton with deformable soft body hands and fingers. When the fingers grasp an object, the contacts generated deform the meshes of the fingers, i.e., change their volumes. Then all internal forces resulting from volume changes are computed. Jain & Liu (2011) simulate fingers in a grasp as soft bodies, but argue that deformations caused by contacts are small and propose simulating such deformations by computing the surface of the deformable body rather than its full volume. Despite this improvement, they must still simulate a subset of vertices surrounding the contact surface for each contact, at each time-step. Verschoor, Lobo & Otaduy (2018) present a method for grasp simulation for real-time dexterous manipulation. Similarly to Jain & Liu (2011), the simulation integrates an articulated skeleton, soft-bodies, and frictional contact. The simulated hand tracks the user's hand motion in real-time while robustly reacting to contacts and collisions with the objects in the virtual environment. However, instead of modeling the contacts as constraints

they instead model them as energies that penalize the distance between each contact point and an artificial anchor point.

Both works involve computations that scale linearly with the number of end-effectors in a grasp. This can cause a performance bottleneck when simulating a grasps with large sets of end-effectors simultaneously. In our method, we choose to model the fingers as rigid bodies rather than soft bodies, which immediately lowers computations to only one contact constraint per contact point. We then further enhance performance by consolidating all contact constraints into just one constraint.

In another work that simulates hand-object interactions in VR for real-time, Höll, Oberweger, Arth & Lepetit (2018) propose starting contact solving before visual collision between the hand and object. The authors define a small threshold distance around the virtual objects and monitor for potential collisions with the hand. Once the distance between the surface of the hand's model and an object is smaller than the threshold, a contact point is created. Then this point is used to compute and apply the different forces.

When modeling contacts, researchers often assume that the contact geometry is a single point. While a contact point is enough to prevent interpenetration, it only models the linear behavior. Multiple contact points are required to model spinning or rolling behavior. In Bouchard *et al.* (2015), the authors propose a new approach to modeling contact by augmenting an individual point-contact with rolling and spinning friction constraints. The result is a 6D constraint that appropriately provides rotational torques to prevent spinning.

Moreover, reducing contacts is another approach used to simulate contact-rich interactions. In Narang *et al.* (2022), the authors use contact grouping, or patching, to reduce the number of contacts. The concept groups a set of contacts that in close vicinity of each other, and that share a normal. These contact patches are then treated as if they're one contact, which the authors argue makes contact generation and solving much quicker.

Similarly, the authors of Talvas, Marchal, Duriez & Otaduy (2015) employ a contact reduction technique to simulate virtual manipulation with fingers modeled as soft-bodies. They use a combination of contact patches and volume constraints to reduce the large number of contacts typically involved in soft-body dynamics.

In an alternative approach, Andrews, Teichmann & Kry (2016) propose an approach that simplifies one interaction to a single 6D mass-spring. They present an incremental algorithm that formulates a multi-body system and its inter-connectivity as a graph, which will then be used to compute the system's dynamics. They achieve this formulation by computing linear maps of body twists as a function of the reduced system state. This results in a collection of linearized models that provide a simulation of the system.

3.2 Grasp synthesis and planning

According to Rosales, Suárez, Gabiccini & Bicchi (2012) "a grasp is a configuration of a hand and an object adjoined at a certain contact points." As such some researchers approached grasp synthesis primarily as an optimization problem by finding the best configuration to achieve a successful grasp that satisfy a desired outcome.

For instance, Rosales *et al.* (2012) proposed solving a grasp problem by modeling physical hand muscles as virtual springs at the joints. These springs guide the hand towards a desired reference grasp pose. The authors define feasibility, and prehensility as criteria to determine the quality of the generated grasp. Feasibility measures whether the each point on each fingertip is in contact at the corresponding regions on the object. Whereas prehensility measures if there's any slippage or object movement and counteracts it by generated joint torques.

Furthermore, Liu (2009) proposed a method that solves for generating hand manipulations on a given object moving along a trajectory, and an initial grasping pose. Contact planning and hand configuration synthesis are the two phases of the proposed algorithm that complete the generation process. In the first phase, contact forces are calculated to achieve object motion within a small window of frames from the input trajectory. If no set of contact forces can

be found to achieve the desired object motion, the author uses an iterative algorithm to add contact points to the initial grasping pose until the desired object motion is achieved. In the second phase of the process, the resulting set of contact forces is used as input. Hand motion synthesis is modelled as an optimization problem that solves for hand poses, torques, and contact positions. One disadvantage of this method is that it requires an initial grasp as input, which limits its generalizability to previously unseen objects and eliminates the possibility of real-time applications.

These issues are the addressed in Oprea *et al.* (2019)'s work. The proposed method creates a grasp through a two-step process that includes object interaction and grasping logic. If an object overlaps with a spherical trigger box on the palm of the hand, it is selected during object interaction. The grasping logic checks if the capsule triggers on each finger tip overlap with the selected object once an object is selected. The movement of the corresponding finger is blocked if there is an overlap. The grasp occurs when all three fingers' movements, the thumb, index, and middle fingers, are blocked. The finger movement is optimised to ensure smooth and plausible animations by removing any sudden movements that differ too much from previous poses. The authors don't say how the object is grasped, but it's assumed that it simply snaps to the hand and follows its movements. It is worth noting that this method is implemented in Unreal Engine, and is suited for VR applications.

Data-driven grasp synthesis methods, in contrast to analytical methods, treat grasp synthesis as a sampling problem, with various grasp configuration parameters sampled from a curated search space. Researchers use less specific parameterization for the grasp, such as using a single approach vector rather than individual fingertip positions, according to the authors Bohg, Morales, Asfour & Kragic (2013), who examine the most popular approaches in this group. As a result, data-driven approaches are more generalizable than analytical approaches, though the resulting grasps are not guaranteed to be stable or at equilibrium.

Based on how much is known about the target grasping object, the authors divide data-driven approaches into three categories.

1. Approaches based on the assumption that the object is well-known. These methods make use of offline datasets that map objects to a set of appropriate grasps. The appropriate grasp is retrieved from the dataset once the robot or virtual character recognises the object.
2. Approaches based on the assumption that the target object is familiar, i.e., similar to but not identical to objects previously encountered. Color, texture, and object type can all be used to describe similarity. The proper grasp is assumed to be the same as the one that corresponds to a similar object.
3. These approaches identify the object structure and features, and retrieve candidate grasps ranked by quality, without assuming any prior knowledge about the object.

The authors further divide these categories into subcategories based on a number of factors that will not be addressed here.

In a method that generates grasping and hand manipulation, Andrews & Kry (2013) exploit the pseudo-cyclic finger motion nature of a set of manipulation tasks such as tuning a dial as a way to speed up the generation process. The generated manipulation occurs over three phases, approach, actuate, and release. During approach, the hand and fingers form a grasping pose, the resulting wrenches from contact forces are used to rotate and/or translate the object during the actuation phase. The object is then released during the releasing phase where fingers and hand assume the next appropriate pose to continue the cycle towards the desired manipulation goal. To achieve these phases, the authors train controllers in a simulated environment using reinforcement learning to create the appropriate control policy.

In real world applications, the problem of grasping is sometimes coupled with path planning for robots. In Stilman, Schamburek, Kuffner & Asfour (2007), the authors tackle the combined problem of manipulation and navigation. In this work, the authors approach the problem of the robot moving obstacles away and navigating the cleared path. They propose an algorithm that addresses the raised dimensionality of the search space of this coupled problem. During path planning, the proposed algorithm performs the following tasks: identify a blocking, grasp the movable obstacle, find a placement for the obstacle, manipulate it, then navigate.

A promising approach that handles multifingered hand-object interactions in real time is proposed by Tian, Wang, Manocha & Zhang (2018). The method generates physically-valid grasps in realtime with no assumptions about the hand motion, or the object shapes. To compute a stable grasp, the authors use machine learning. Firstly, the configuration space is randomly sampled and categorized as in-collision or collision-free samples that determine if the hand and object collide. The collision-free samples are taken as vectors and used as a starting configuration to search for the stable grasps. Then, random velocities are assigned to the collision-free vectors forming a potential grasping configuration. For each configuration a stable grasp is computed.

CHAPTER 4

METHODOLOGY

Starting from a state where a grasp has not yet been formed, we monitor all of the contacts between the end effectors and target object at each time step. These contact forces are transformed to wrenches in the local space of the target object and used to compute a grasp quality based on the force closure metric. Once a sufficient level of quality is detected, the end effectors are merged to a single composite rigid body and a 6D constraint is created between the composite body and the target object that approximates the original grasp. Then, the simulation continues to monitor the contacts between the hand and object, but without solving the dynamical equations. When the grasp quality once again drops below a certain threshold, the end effector bodies are unmerged and then the constraint is disabled. However, while the grasping constraint is active, it allows the object to slide in the hand/gripper and realistic finger motions to occur.

Figure 4.1, shows a summary of all the steps in our approach. The method can be broken down to two main states, ungrasped and grasped. Figure 4.3 shows the MBS in the two states. From here on out, we use the terms, object, rigid body, and part interchangeably.

In the remainder of this section, I will use bold text to emphasize the words that correpsond to the blocks shown in 4.1. At the start of the **ungrasped** state, we begin with an articulated **hand** at rest pose, a corresponding predefined **grasp pose** and a **target object** with a set of **goal points**. We **track** (transition) the hand's pose to the grasp's pose using forward kinematics once. Then we use inverse kinematics to track (**IK track**) the **fingertips** to the predefined **goal points** on the target geometry surface. At this stage, the fingertips are in contact with the object and contact constraints are generated by the physics engine. In the **grasped** state, we **compute the grasp quality Q** in real-time. When a threshold **grasp quality $\mathbf{Q_{Th}}$** value is reached, we disable all contact constraints generated between the fingertips and target object and replace them with one 6D constraint. The constraint is already attached to the target object and missing the second attachment. We merge the fingertips together as one **merged part** (rigid body) and **attach** it to the 6D constraint. We set the primary and secondary axes of the constraint to align with the

axes where slipping/sliding will occur based on the scenario. The axes are set relative to the object's frame of reference located at its center of mass. We **relax the constraint** along these axes to mimic dexterity and keep monitoring the grasp quality. When the quality drops below a minimal value $\mathbf{Q}_M$ we **unmerge** and **detach** the fingertips from the constraint.

Figure 4.2 shows a multi-body system consisting of a target object and fingertips as end-effectors, in a physics simulation. The fingertips, shown as small yellow spheres, on the 3rd segment of each finger, represent the exact position of the end-effectors that the IK solver will later use to compute the distance error to the goal touch points. The goal touch points are shown as small yellow spheres on the target object shown as a big light pink sphere.

In the sections below, we describe in further detail the main contributions of our methodology.

4.1 Grasping

Once simulation begins, we use inverse kinematics to control the hand such that the fingertips move toward the goal points on the target surface. Once the fingertips reach their target positions, they exert contact forces on the object.

We use a process that's packaged with the Vortex SDK to compute grasping quality. The IK solver from the previous step improves on the grasp quality in real-time by continuing to track the goal points. This results in increasing the magnitudes of the contact forces applied on the target object till a minimum quality threshold $\mathbf{Q_{Th}}$ is met. The quality threshold is set heuristically based on the scenario. For each scenario, we run the simulation and record $\mathbf{Q_{Th}}$ as the minimum grasp quality value required for the object to remain constrained at rest, without slipping, with gravity applied. The minimum threshold changes due to the target object's shape and mass, the grasp pose, and which axis the object can slide realistically without losing the grasp.

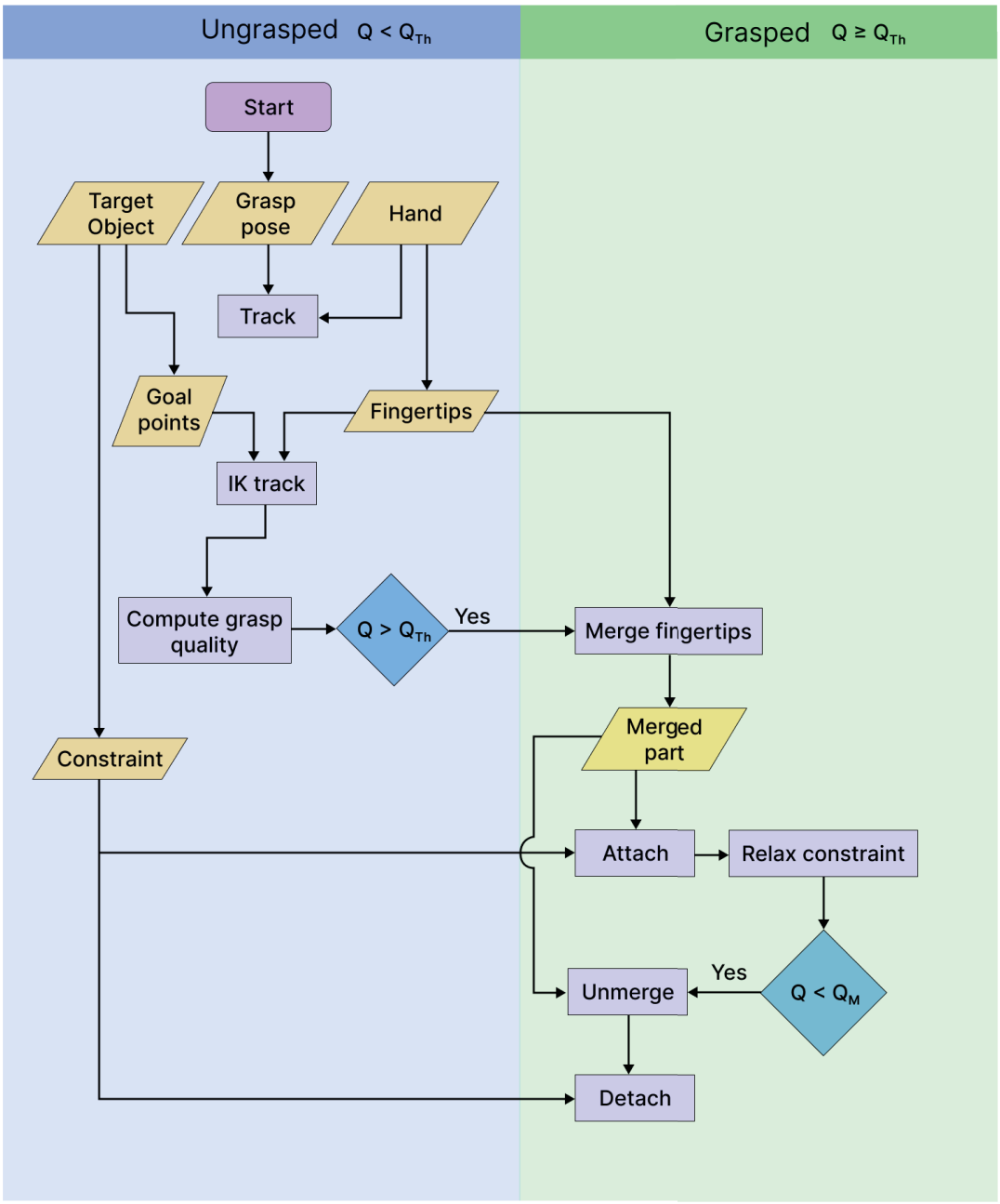


Figure 4.1 Flowchart of our approach

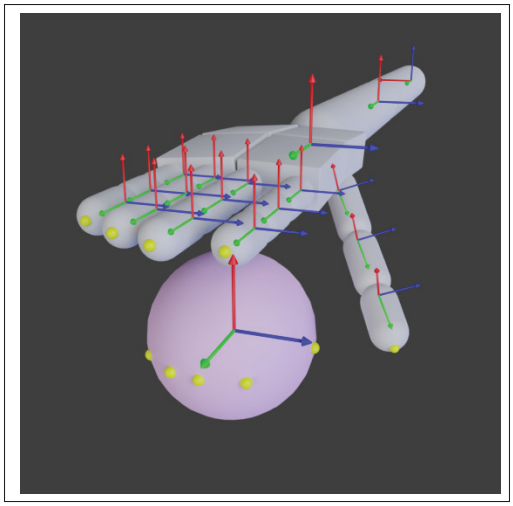
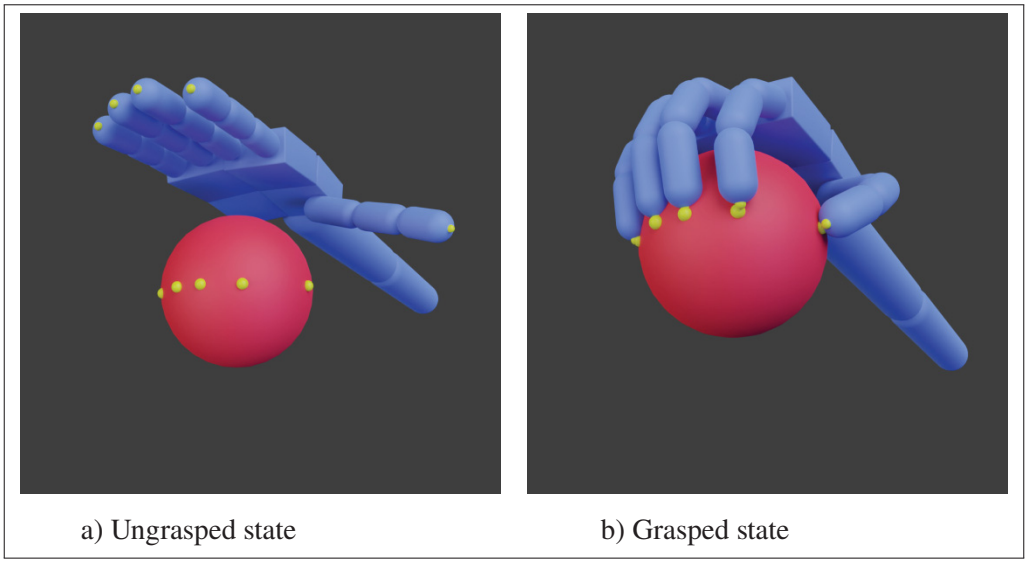


Figure 4.2 MBS consisting of arm, hand (white) and a target object (light pink). Small yellow spheres on the fingertips to specify the end-effectors exact positions. Small yellow spheres on the target object specify the goal touch points. Local joints axes are shown as the three arrows representing axes x, y, z colored red, green, and blue respectively



a) Ungrasped state b) Grasped state

Figure 4.3 The MBS in ungrasped and grasped states

4.2 Contact Approximation

When the quality threshold $\mathbf{Q_{Th}}$ is met, we disable all contact constraints created between the fingertips and the target object and replace them by one 6D constraint located at the center of mass of target object. A constraint in Vortex SDK allows a maximum of two bodies as inputs to couple their movements. These two inputs are referred to as attachments. In our experiments, we add a 6D constraint at the center of mass of the target object, and set the target object as one attachment. At this point, all the fingertips are separate bodies, but only one attachment slot is available. We opt to merge all the fingertips to a single part. Essentially, this treats all fingertips as one rigid body, and the target object as another, which meets the constraint attachments requirement. Vortex allows to seamlessly merge and un-merge rigid-bodies when necessary without breaking the system. Although this would firmly hold the object in the grip of the fingers, dexterity is no longer possible. So we introduce relaxing the constraint on specific axes to allow for sliding, and flexing.

Dexterity To mimic dexterity, frictional behavior must be apparent visually. In other words, the object should be able to slide and rotate relative to fingertips. Furthermore, the fingertips must flex when moving or pulling the object.

When grasping with contact constraints enabled, the object's sliding behavior is proportional to the sum of contact forces applied on the object's surface.

To emulate sliding, we propose setting the 6D joint as a constraint with limits on the Lagrange variables. The constraint limits enforcing the relative motion between the end-effectors and target object are updated as follows

Linear motion

$$\mathbf{F}_{maxX,Y,Z} = \mu \mathbf{F}^{\perp} \tag{4.1}$$

Angular motion

$$\mathbf{F}_{\theta,\phi,\omega} = \mu \|r\| \mathbf{F}^{\perp} \tag{4.2}$$

The vector r represents a moment arm computed by averaging the distances from each contact location to the geometric centre of all N contacts. $\mathbf{F}^{\perp}$ is a scalar value evaluated as

$$\mathbf{F}^{\perp} = \sum_{i}^{N} f_i^T n_i \tag{4.3}$$

where f_i and n_i are the i_{th} contact force and surface normal, respectively.

In the next chapter, we present the two experiments we designed to test our methodology and provide comparisons of the visual appearance and performance between our approach and the ground truth.

CHAPTER 5

RESULTS

In this section, we describe the experiments we devised to test our methodology and compare it to existing methodologies. This thesis work was done using CMLabs Vortex Studio; a real-time physics simulation engine and visualization software, as well as using their C++ SDK which allows for a low-level access to the engine's functionality.

5.1 Experiments

We devised the following two experiments:

- A hand with all five fingers clutching a ball.
- A hand with a two-finger pinch grip on a thin rectangular item representing a phone.

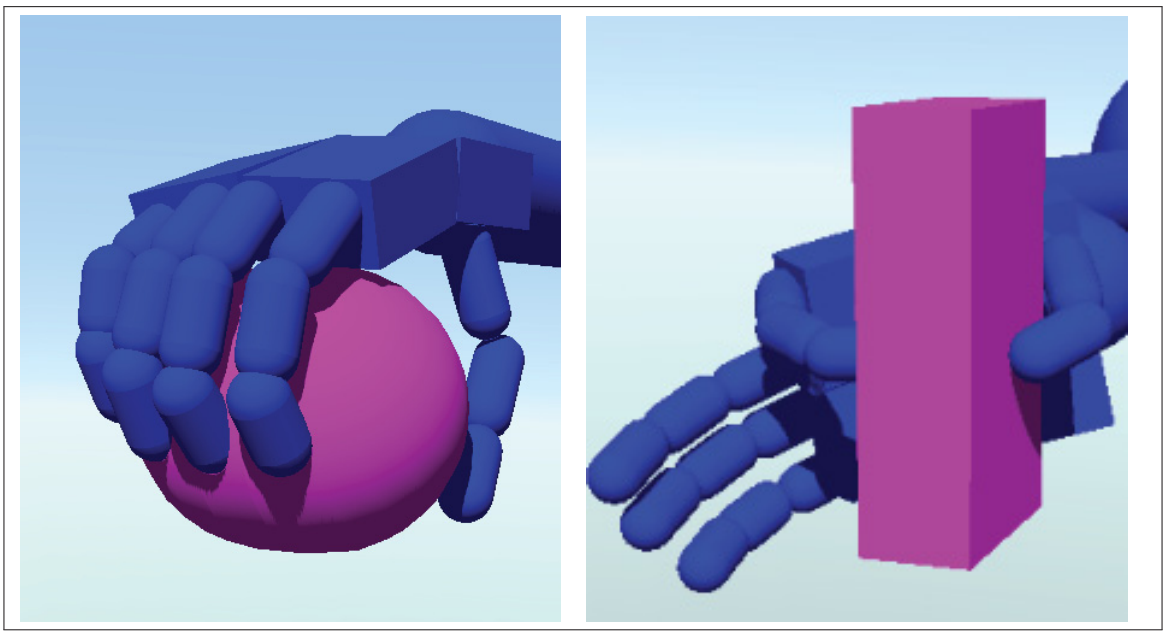
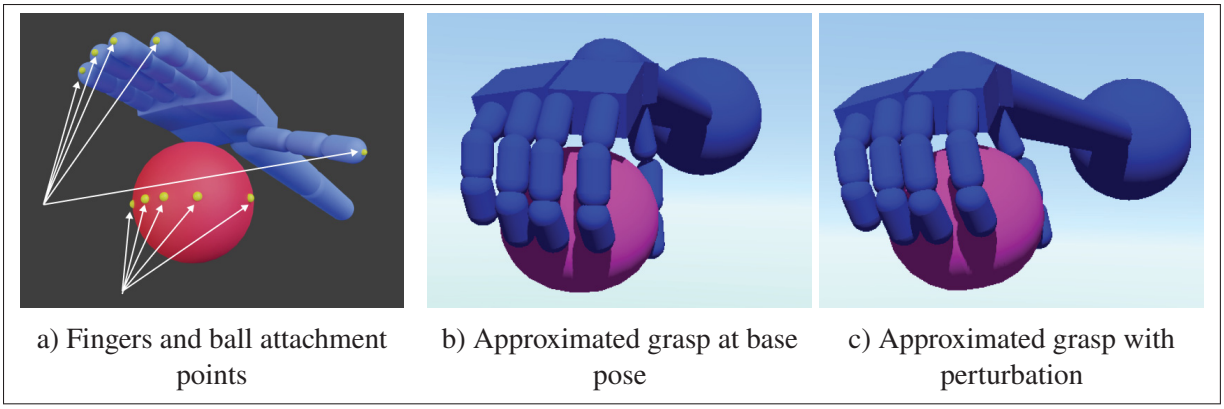


Figure 5.1 Testing scenarios for performance evaluation

To carry out both experiments, we create two multi-body systems. We build a three-dimensional humanoid with 63 degrees of freedom. Based on de Leva (1996), the mass and geometry are chosen to correspond to an average adult male. The total mass of the arm is 5.45 kg. Each finger



a) Fingers and ball attachment points

b) Approximated grasp at base pose

c) Approximated grasp with perturbation

Figure 5.2 Experiment 1 configuration

weighs 60 g with 20 g for each segment. The palm of the hand is 250 g. Finger joints have a stiffness of 0.250 $N.m/rad$ and a damping of 0.05 kg.m^2/s These values are heuristically chosen to induce a grasp that is stable, but not too stiff to allow for flexing.

For each experiment, we predefined poses for each MBS, then used the IK solver to set the MBS to the poses in each experiment as shown in Figure 5.1. Furthermore, we decided to use force closure types of grasps.

The number of end-effectors, i.e., fingertips used in each experiment is determined by the target object to be grasped. We selected the number of fingertips for each experiment that is high enough to achieve the force closure grasp efficiently but low enough to retain performance. For scenario 1 and 2, the number of end effectors is 5 and 2 respectively corresponding to the number of fingers involved in the scenario. Furthermore, we defined a set of target contact points for each target object, as detailed below.

5.1.1 Experiment 1: A hand with all five fingers clutching a ball

Clutch grasping a ball can model and generalize to a lot of real world objects we interact with, so it is our first choice as an experiment.

The hand in this scenario is fully modeled. As an approximation for each finger segment, we used a capsule as collision geometry. We modeled the collision geometry of the palm of the

hand by combining two boxes. At the third segment of each finger, we placed an attachment point to represent the end-effector as shown in Figure 5.2a The target object is a ball of 200 g mass. We placed five attachment points on the ball, and positioned them to be close to where the fingertips are when the hand is in a clutch pose.

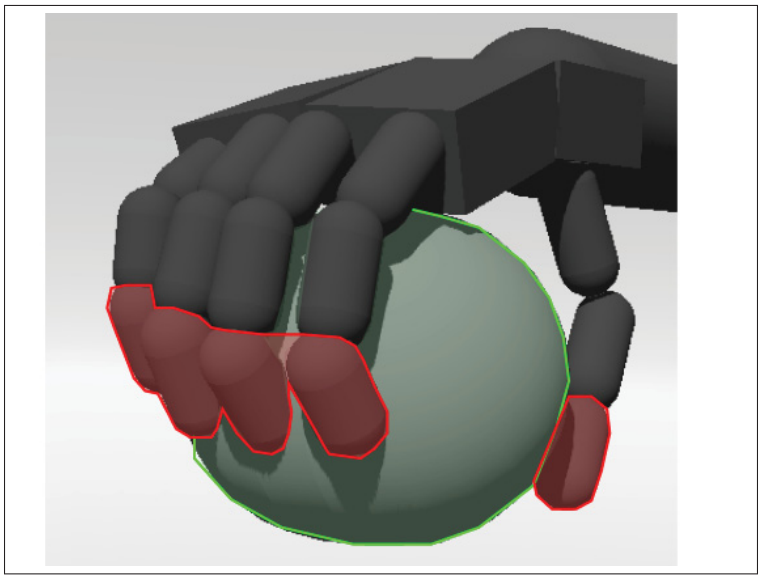


Figure 5.3 Experiment 1, constraint attachments. Attachment 1: target object (green highlight). Attachment 2: merged fingertips (red highlight)

It was necessary to change the grasping pose so that the fingertips were almost in contact with the ball. The IK controller will then push the fingertips into touch and provide sufficient force to produce a force closure. If the fingers do not exert enough force, the grasp quality threshold is not met, and no approximation occurs. Furthermore, for the grasp quality to be computed, Vortex SDK requires contacts between the hand and the object to be deactivated, which introduces intersections between fingertips and the ball's collision geometry, which is not realistic.

To demonstrate sliding behavior, we apply two forces as perturbations, shown in Figure 5.4. First, we apply a force $\mathbf{f}_{p_1} = (0, -9.5, 0)$ N, then reset the simulation and apply force $\mathbf{f}_{p_2} = (0, 0, -2.5)$ N and observe the behavior. Figure 5.2b depicts the grip approximation's base pose before applying

force $\mathbf{f}_{p_1}$, and Figure 5.2c depicts the fingers and hand flexing to react to the ball moving while $\mathbf{f}_{p_1}$ without losing the grasp. This is better demonstrated in the supplementary video.

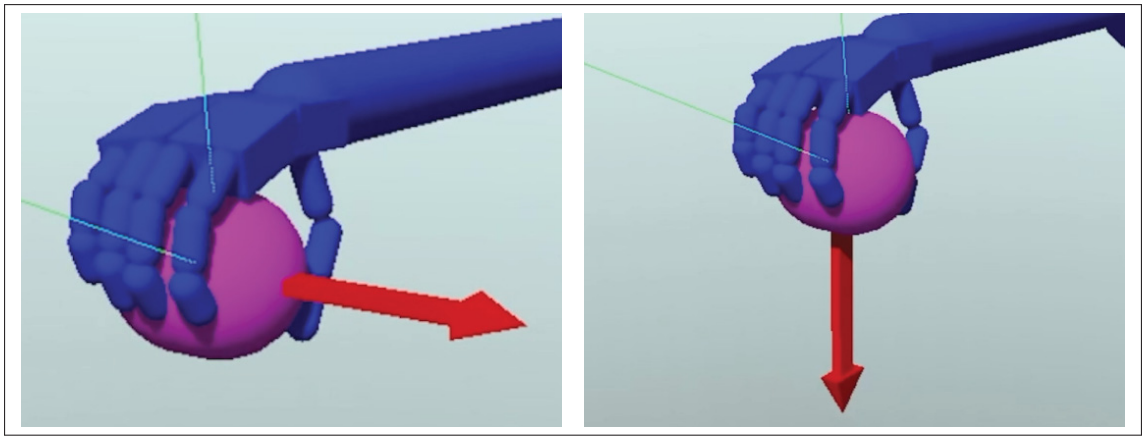


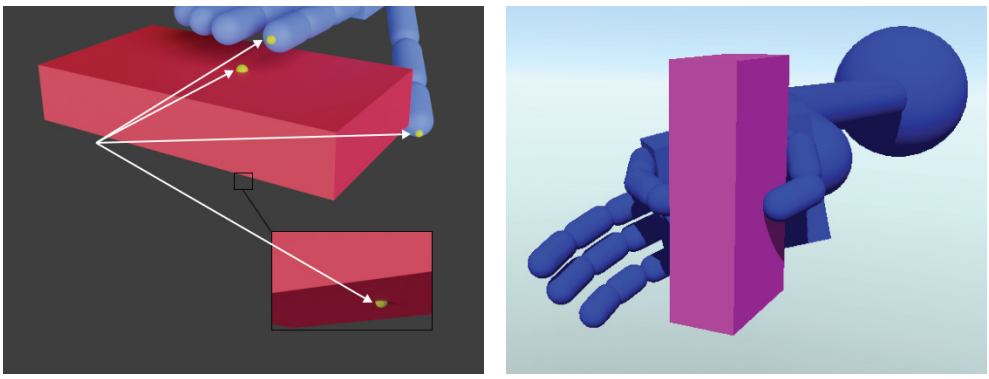
Figure 5.4 Experiment 1 perturbation forces $\mathbf{f}_{p_1}$ (left), and $\mathbf{f}_{p_2}$(right)

Approximation works in this scenario, with one exception, and the results are similar to the ground truth, i.e., non-approximated simulation with a few key differences. First, there's more sliding and dynamics in the full simulation, as compared to our approach. Furthermore, the fingertips move as on object, which may standout as a visual flaw. Another distinction is the sliding time; when utilising our method, sliding might begin earlier. Finally, there are penetration errors between the fingers and the object that do not appear in the full simulation.

5.1.2 Experiment 2: A hand with a two-finger pinch grip on a thin rectangular item

Pinch grasping a light object can demonstrate a grasp's stability and accuracy. The hand in this scenario is the same as the previous scenario. The target object is a rectangular object resembling a phone of 100 g mass. We placed two attachment points on the phone, and positioned them to be close to where the fingertips are when the hand is in a pinch pose.

The difficulty with this scenario is higher than the previous in terms of setup. It is harder now because there are only two fingers involved to generate contact forces.



a) Fingers and phone attachment points b) Approximated pinch at base pose

Figure 5.5 Experiment 2 configuration

Similar to the previous scenario, we create two perturbations, by applying two forces $\mathbf{f}_{p_1} = (-2.0, 0, 0)$ and $\mathbf{f}_{p_2} = (0, 0, -0.5)$ to the phone's center of mass as shown in Figure 5.6. In Figure 5.5b we show the pinch approximation's base pose before perturbation. There's very minimal flexing before the object slides off and the grasp is lost.

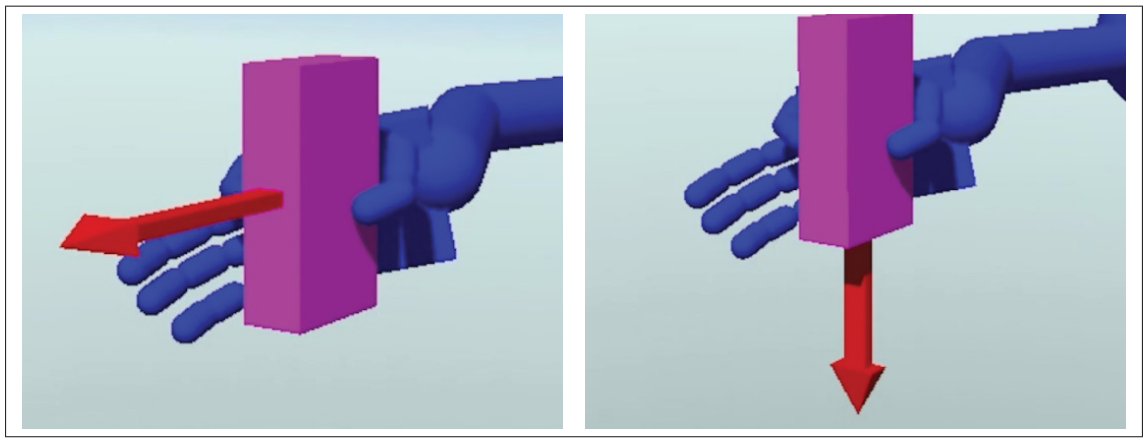


Figure 5.6 Experiment 2 perturbation forces $\mathbf{f}_{p_1}$ (left), and $\mathbf{f}_{p_2}$(right)

Approximation works well in this scenario as well, with the same problem if overlapping/intersecting geometry. It is less noticeable in this case. However, it is important to point out that the chosen target points on the phone are positioned as close as possible to its center of mass.

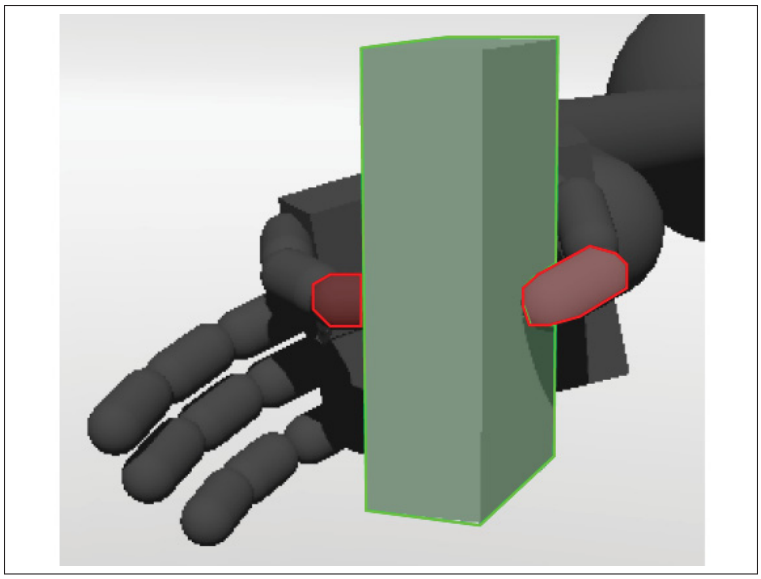


Figure 5.7 Experiment 2, constraint attachments. Attachment 1: target object (green highlight). Attachment 2: merged fingertips (red highlight)

5.2 Performance

To assess the performance enhancements of our method, we measured the time it takes to solve for all contact constraints of the grasp during one simulation time-step, and compared it to the ground truth. These measurements are done while running the simulation on a 2.70-GHz Intel Core i7-6820HK double-threaded 4-core CPU. The results are shown in Table 5.1.

Table 5.1 Comparing the performance of the baseline simulation versus our approach.

Scenario	**Full simulation** (IK + dynamics)	**Our approach**
Experiment 1	4400μs (4306μs + 94μs)	12μs
Experiment 2	2300μs (2261μs + 39μs)	5μs

It is important to note, the full simulation requires the IK solve, which takes the longest to compute. Our approach removes the need for IK during the grasp. This already improves the

performance a lot. Furthermore, when comparing timing for solving the dynamics only, our method simulates a grasp visually comparable to ground truth 7 times faster.

CONCLUSION AND RECOMMENDATIONS

Ultimately, the success of future virtual characters will be determined by the development of algorithms that allow them to not only sense their surroundings, but also effectively govern their interaction with items. It takes a lot of processing power to simulate such interactions in real time. In this thesis, we show how our technique can reduce the performance impact of modeling interactions and grasps while maintaining visual integrity. Each time step of the simulation, the ground truth simulation of a grasp of an object entails computing and updating the various contact restrictions and their forces independently. Our technique effectively minimizes that overhead by replacing all contact requirements with a single 6D constraint, allowing the grasping simulation to retain the visual appearance of the ground truth while incurring a significantly lower performance cost.

While our approach has significantly improved performance, there are a few issues worth mentioning. To build and test our method, we used CMLabs Vortex SDK as our physics simulation engine of choice. The SDK has a function for calculating grasp quality, but doing so disables all contacts between the objects involved in the grasp. This caused a problem of implausible collision geometries intersections, such as fingertips penetrating through a solid object. In future work, this problem can be mitigated by introducing layered collision geometries for an object. For example, for a spherical ball, introduce two spherical collision geometries, an inner one with a small radius that matches the ball, and an outer one with a slightly larger radius, so that when the grasp quality computation function is used, the fingertips will only intersect with the outer collision sphere. Another solution is to alter the grasp computation function to limit the interpenetration allowed.

Another current shortcoming of our approach is that when all end-effectors are merged into a single rigid-body, individual end-effector motion is lost. This may result in visual problems during simulation. For example, if a grasped object slides away and loses contact with one or

more fingertips, the fingers can continue flex as if the object is still in contact with them. This problem can be avoided in the future by implementing a custom constraint that allows for more than two attachments.

Finally, our technique provides a significant performance improvement with negligible accuracy penalties. Positioning it as a suitable performance optimization strategy for real-time simulation applications, particularly video games on constrained hardware platforms such as consoles, or where high performance is critical, such as virtual-reality and augmented-reality experiences.

BIBLIOGRAPHY

Andrews, S. & Erleben, K. (2021). Contact and friction simulation for computer graphics. In *ACM SIGGRAPH 2021 Courses*. doi: 10.1145/3450508.3464571.

Andrews, S. & Kry, P. G. (2013). Goal directed multi-finger manipulation: Control policies and analysis. *Computers & Graphics*, 37(7). doi: 10.1016/j.cag.2013.04.007.

Andrews, S., Teichmann, M. & Kry, P. G. (2016). Blended Linear Models for Reduced Compliant Mechanical Systems. *IEEE Transactions on Visualization and Computer Graphics*, 22(3), 1209-1222. doi: 10.1109/TVCG.2015.2453951.

Bohg, J., Morales, A., Asfour, T. & Kragic, D. (2013). Data-driven grasp synthesis—a survey. *IEEE Trans. on Robotics*, 30(2). doi: 10.1109/TRO.2013.2289018.

Bouchard, C., Nesme, M., Tournier, M., Wang, B., Faure, F. & Kry, P. (2015). 6D frictional contact for rigid bodies. *Proceedings of Graphics Interface 2015*, (GI 2015), 105–114. doi: 10.20380/GI2015.14.

de Leva, P. (1996). Adjustments to Zatsiorsky-Seluyanov's segment inertia parameters. *Journal of Biomechanics*, 29(9). doi: 10.1016/0021-9290(95)00178-6.

Goldstein, H., Poole, C. & Safko, J. (2002). Classical mechanics. American Association of Physics Teachers.

Höll, M., Oberweger, M., Arth, C. & Lepetit, V. (2018). Efficient Physics-Based Implementation for Realistic Hand-Object Interaction in Virtual Reality. *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. doi: 10.1109/VR.2018.8448284.

Howard, W. S. & Kumar, V. (1996). On the stability of grasped objects. *IEEE Trans. on Robotics and Automation*, 12(6). doi: 10.1109/70.544773.

Jain, S. & Liu, C. K. (2011). Controlling Physics-Based Characters Using Soft Contacts. *ACM Trans. on Graphics*, 30(6). doi: 10.1145/2070781.2024197.

Liu, C. K. (2009). Dextrous Manipulation from a Grasping Pose. *ACM Trans. on Graphics*, 28(3). doi: 10.1145/1531326.1531365.

Narang, Y., Storey, K., Akinola, I., Macklin, M., Reist, P., Wawrzyniak, L., Guo, Y., Moravanszky, A., State, G., Lu, M. et al. (2022). Factory: Fast Contact for Robotic Assembly. doi: 10.48550/arXiv.2205.03532.

Oprea, S., Martinez-Gonzalez, P., Garcia-Garcia, A., Castro-Vargas, J. A., Orts-Escolano, S. & Garcia-Rodriguez, J. (2019). A visually realistic grasping system for object manipulation and interaction in virtual reality environments. *Computers & Graphics*, 83, 77-86. doi: 10.1016/j.cag.2019.07.003.

Rosales, C., Suárez, R., Gabiccini, M. & Bicchi, A. (2012). On the synthesis of feasible and prehensile robotic grasps. *IEEE International Conference on Robotics and Automation*, (ICRA 2012), 550-556. doi: 10.1109/ICRA.2012.6225238.

Stilman, M., Schamburek, J.-U., Kuffner, J. & Asfour, T. (2007). Manipulation planning among movable obstacles. *Proceedings 2007 IEEE international conference on robotics and automation*. doi: 10.1109/ROBOT.2007.363986.

Talvas, A., Marchal, M., Duriez, C. & Otaduy, M. A. (2015). Aggregate constraints for virtual manipulation with soft fingers. *IEEE Trans. on Visualization and Computer Graphics*, 21(4). doi: 10.1109/TVCG.2015.2391863.

Tian, H., Wang, C., Manocha, D. & Zhang, X. (2018). Realtime hand-object interaction using learned grasp space for virtual environments. *IEEE Trans. on Visualization and Computer Graphics*, 25(8). doi: 10.1109/TVCG.2018.2849381.

Verschoor, M., Lobo, D. & Otaduy, M. A. (2018). Soft hand simulation for smooth and robust natural interaction. *2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*. doi: 10.1109/VR.2018.8447555.