Efficient and Transparent Haptic Rendering of Rigid Body Motion with Constraints

by

Daniela Constantinescu

B.Sc. (Mechanical Engineering), University of Brasov, 1986

M.A.Sc. (Mechanical Engineering), University of British Columbia, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

THE UNIVERSITY OF BRITISH COLUMBIA

December 2004

Abstract

This work presents general purpose simulation and control techniques for efficient and transparent haptic rendering of rigid body motion with constraints. Transparent interaction is achieved by enabling users to feel collisions and to manipulate both virtual objects and virtual linkages. Efficient rendering is accomplished through fast approximations of rigid body interaction implemented in a local model with haptic performance.

Hapic rendering of impacts is based on a new representation of rigid body contact. In this representation, contacts are infinitely stiff when they arise and have limited stiffness thereafter. Multiple impacts are resolved simultaneously, in a manner consistent with conservation of energy principles and with the force capabilities of the haptic device. Haptic rendering of impacts is beneficial in training simulators for dental procedures and bone surgeries, as well as CAD and virtual prototyping systems with force feedback.

Realistic linkage manipulation is enabled by permitting users to operate linkages from any link and through singularities while restricting their motion according to the virtual environment geometry and the linkage topology. Linkage topology is imposed on users through penalizing users' departure from the configuration manifold of the virtual linkage. Operation of links with insufficient degrees of freedom is important in applications like training for laparoscopy, where the scope limits the tool motion at the entry point.

Efficient rendering of rigid body motion with constraints is enabled by interfacing the device to a simulation through a local model of interaction. The model comprises constraints imposed on the virtual tool by virtual objects within an ϵ-active neighborhood of the virtual tool and a dynamic proxy of the virtual tool. This model is the first that can be used to constrain both the translation and the rotation of the device and to add realistic forces to virtual environments generated using any commercial simulation package with interactive performance. The model is beneficial in consumer-grade haptic applications, because it allows easy development of haptic applications by users without detailed haptic knowledge. It can also be used to enable cooperative haptic manipulations in applications that involve two-handed operations and/or multiple users.

Table of Contents

Abstract ii

Table of Contents iii

List of Tables vi

List of Figures vii

Glossary xii

Acknowledgements xv

1 Introduction 1

1.1 Objective 4

1.2 Contributions 4

1.3 Organization of the thesis 7

2 Physically-based rigid multibody simulations 10

2.1 Interactive simulations 10

2.1.1 Coordinates for rigid multibody dynamics 13

2.1.2 Dynamic models of rigid multibody systems 14

2.1.2.1 Penalty-based simulations 15

2.1.2.2 Constraint-based simulations 16

2.1.2.3 Impulse-based simulations 18

2.2 Impact simulation 19

2.3 Haptic interaction within rigid multibody virtual environments 22

2.3.1 Haptic controllers 23

2.3.2 Local models of interaction 24

2.3.3 Haptic rendering of impact 25

2.3.4 Haptic rendering of friction 26

2.3.5 Summary 26

3 Experimental setup 28

4 Haptic rendering of rigid contact 32

4.1 Synopsis of the impulse-augmented penalty simulation and of the haptic controller 33

4.2 The contact model 35

4.3 Resting contact 37

4.3.1 Friction modeling 39

4.4 Colliding contact 46

4.4.1 Passive collision resolution 48

4.4.1.1 Independent constraints 49

4.4.1.2 Overdetermined constraints 52

4.4.2 Rendering collisions to users 54

4.5 The performance of the impulse-augmented penalty simulation 54

4.5.1 Best performance 56

4.5.2 Accounting for device limitations 57

4.5.3 Experimental validation 65

4.6 Discussion 69

5 Manipulation of serial linkages 70

5.1 Control architecture for realistic linkage manipulation 71

5.2 Linkage contacts at user's hand 72

5.3 Haptic rendering of linkage inertia 74

5.4 Haptic rendering of linkage topology 76

5.4.1 Simulations 78

5.4.2 Experiments 81

6 Efficient haptic rendering of rigid body motion with constraints 85

6.1 Synopsis of the local model 86

6.2 Local geometry 87

6.2.1 Active geometry 87

6.2.2 Local proxy deformation 91

6.2.3 ϵ-active local geometry 92

6.2.4 Contact clustering and constraint coherence 95

6.3 Local dynamics 98

6.3.1 Local contact interactions applied to users 98

6.3.2 Local control 99

6.4 Experiments 100

6.4.1 Passive collision dynamics in the impulse-augmented penalty local model 100

6.4.2 Transparency in the local model of interaction 102

6.4.3 The effect of the teleoperation controller on the achievable contact stiffness 108
6.4.4 Dynamic contact 109

7 Conclusion 112
7.1 Summary of Contributions 112
7.2 Future Work 114

Bibliography 117

A Proof of implicit elimination of constraint overdeterminancy 126

B Weighted generalized inverses for linkage manipulation 128
B.1 Weighted generalized inverses of matrices 128
B.2 Weighted generalized inverses applied to robotics 129

C Equality of null spaces of $\mathbf{\Lambda}_h^{-1}$ and $\mathbf{J}_h^T$ 132

D The dynamics and control of the planar haptic interface 134
D.1 Interface actuation and dynamics 134
D.2 Interface control 136

List of Tables

4.1 Parameters of the teleoperation controller and the unilateral coupler connecting the virtual environment to the haptic device. 56

5.1 Parameters of the three links planar virtual linkage operated by the user in the simulations. . . 78

List of Figures

1.1 The key components of a haptic interaction system and their conflicting requirements for stability and realism. 2

1.2 Alleviation of the effect of the computational delay and the geometry-dependent environment stiffness via control. 3

1.3 Elimination of the computational delay of the simulation via new collision detection algorithms. 3

1.4 Mitigation of the effect of the computational delay of the simulation via a local model of interaction. 4

1.5 The contact model the enables users to feel collisions: contacts have infinite stiffness upon contact and finite stiffness during contact. Users feel impulsive forces upon contact and penalty and friction forces during contact. 5

1.6 Users perceive topological constraints through penalties applied to the device when their hand leaves the configuration manifold of the linkage (vh is the position and orientation of the user's hand on the virtual linkage, while h is their real position and orientation). 6

1.7 The local model of rigid body interaction includes a dynamic proxy of the virtual tool and active and predicted motion constraints imposed on the virtual tool by nearby objects. 7

2.1 The simulation loop and the scope of the survey: prior work in dynamic response. 11

2.2 Coordinates used to describe the system state in rigid multibody simulations. 13

2.3 The contact states in a penalty-based rigid multibody simulation. 15

2.4 A rectangular object interacting with a virtual wall. Each surface contact is represented through two point contacts. 16

2.5 The contact states in a constraint-based rigid multibody simulation. 17

2.6 The corner law. 17

2.7 The contact states in an impulse-based rigid multibody simulation. 18

2.8 The virtual coupler. 23

3.1 The planar haptic simulation system used for experiments throughout this thesis. 28

3.2 Communication between the virtual environment simulation and the force control loop in the planar haptic interaction system used in this thesis. 30

3.3 Testbed virtual environments used in the experiments. 31

4.1 Haptic manipulation in an impulse-augmented penalty virtual world. 33

4.2 The impulse-augmented penalty-based rigid body contact model: contact rigidity is exactly enforced upon contact and only approximately enforced during contact. 35

4.3 Contact information received from collision detection. 35

4.4 Example transition from free motion to resting contact during a fixed step haptic simulation. $\boldsymbol{\omega}$ is the angular velocity of the body, $\mathbf{v}$ is the velocity of its center of mass, $\mathbf{v}_A$ is the velocity of point A on the body, and t is the time step of the simulation. 36

4.5 Example contact forces and and hand wrenches arising during the haptic manipulation of a contact group with dynamics computed in configuration space. 38

4.6 Simulated peg-in-hole manipulation. 40

4.7 Peg-in-hole simulation. Friction is simulated using the reset-integrator model [61] with various damping coefficients. 41

4.8 Classical approximation of Coulomb friction. 43

4.9 Peg-in-hole simulation. Friction is simulated using the critically damped reset-integrator model [61], the model proposed by Nahvi *et al.* [113], the modified Dahl model [67], and the classical dry friction approximation. 44

4.10 Forces felt by users and COM trajectories of a virtual peg sliding with friction in a tight hole due to a sinusoidal force applied to its COM. Various friction models are used to simulate dry friction. Note that all models render slip and stick. Perceivable oscillations occur during stiction when friction is simulated using the modified Dahl model [67] (Figure 4.10(b)) and the model based on human finger-pad characteristics [113] (Figure 4.10(c)). Some slip to stick transitions are missed by the model based on the human finger-pad characteristics [113] (Figure 4.10(c)). . 45

4.11 Two link planar manipulator whose loss of kinetic energy for various values of the coefficient of restitution is shown in Figure 4.12 for two contact geometries. The simplifying assumption in Equation (4.24) imposes the particular solution shown in Figure 4.12(b). 51

4.12 Loss of kinetic energy of the planar two link manipulator in Figure 4.11 during one frictionless collision for two contact geometries. 51

4.13 Planar virtual world used in simulations and experiments. Starting from rest and the position shown, the rectangular object is pushed into the lower right corner by a controlled constant force. 55

4.14 1 DOF mechanical equivalents of the controllers connecting the haptic device and the virtual environment. Controller parameters are given in Table 4.1. 56

4.15 Simulated hand trajectories obtained when constraint-based ("ideal"), impulse-augmented penalty-based ("IAPB"), and penalty-based ("PB") interactions are applied to users by the four channel controller, and when constraint-based interactions are applied by the unilateral coupler ("CB$_{uc}$"). The device applies the simulated impulses to users in one step. 58

4.16 Simulated forces along the x-axis and torques applied to users by the four channel controller when the virtual world is generated using the constraint-based (“ideal”), the impulse-augmented penalty-based (“IAPB”), and the penalty-based (“PB”) methods, and applied by the unilateral coupler when the virtual environment is generated using the constraint-based method (“CB$_{uc}$”). The device applies the simulated impulses to users in one step. 59

4.17 Kinetic energy of the users’ hand when constraint-based (“ideal”), impulse-augmented penalty-based (“IAPB”), and penalty-based (“PB”) interactions are appllied to users by the four channel controller, and when constraint-based interactions are applied by the unilateral coupler (“CB$_{uc}$”). The device applies the simulated impulses to users in one step. 60

4.18 Simulated hand trajectories during user interaction with the impulse-augmented penalty-based virtual world when the device limitations are ignored and when they are taken into account (“ideal” - full constraint-based collision impulses are applied to users; “p$_{full}$” - full collision impulses are applied to users; “p$_{saturated}$” - collision impulses are saturated on the device; “p$_{scaled}$” - collision impulses are scaled in the virtual environment; “p$_{spread}$” - collision impulses are spread over several simulation steps when necessary). 62

4.19 Simulated hand trajectories obtained when constraint-based (“ideal”), impulse-augmented penalty-based (“IAPB LM”), and penalty-based (“PB”) interactions are transmitted to users by a four channel controller, and when constraint-based interactions are transmitted to users by a unilateral coupler (“CB$_{uc}$”). Collision impulses computed using the proposed method are spread over several simulation steps when necessary. 63

4.20 Simulated forces and torques applied to users by the four channel controller when the virtual world is generated using the constraint-based (“ideal”), the impulse-augmented penalty-based (“IAPB LM”) and the penalty-based (“PB”) methods, and applied by the unilateral coupler when the virtual environment is generated using the constraint-based method (“CB$_{uc}$)”. Collision impulses computed using the proposed technique are spread over several simulation steps when necessary. 64

4.21 Kinetic energy of the users’ hand when constraint-based (“ideal”), impulse-augmented penalty-based (“IAPB LM”), and penalty-based (“PB”) interactions are applied to users by the four channel controller, and when constraint-based interactions are applied by the unilateral coupler (“CB$_{uc}$”). Collision impulses computed using the proposed method are spread over several simulation steps when necessary. 65

4.22 Device trajectories when the peg is pushed into the corner by a controller force. The virtual environment is generated using the impulse-augmented penalty-based and the penalty approaches (“IAPB” - impulse-augmented penalty-based world, full collision impulses applied to the device; “IAPB LM” - impulse-augmented penalty-based world, limited impulses applied to the device; “PB” - penalty world). The “ideal” trajectory is generated using a constraint-based simulation. 67

4.23 Environment wrenches applied to the device when the peg is pushed into the corner by a controlled force. The virtual environment is generated using the impulse-augmented penalty-based and the penalty approaches ("IAPB" - impulse-augmented penalty-based world, full collision impulses applied to the device; "IAPB LM" - impulse-augmented penalty-based world, limited impulses applied to the device; "PB" - penalty world). The "ideal" wrenches are generated using a constraint-based simulation. 68

4.24 Kinetic energy of the device while it is pushed towards the corner by a controlled force. The virtual environment is generated using the impulse-augmented penalty-based and the penalty approaches ("IAPB" - impulse-augmented penalty-based world, full collision impulses applied to the device; "IAPB LM" - impulse-augmented penalty-based world, limited impulses applied to the device; "PB" - penalty world). The "ideal" wrenches are generated using a constraint-based simulation. 69

5.1 Realistic haptic manipulation of virtual linkages in an impulse-augmented penalty virtual world. 71

5.2 1 DOF mechanical equivalent of the control architecture. 72

5.3 Example manipulations restricted by the topology of the virtual linkage. 76

5.4 Penalty wrench constraining users to the configuration manifold of the virtual linkage that they manipulate. 77

5.5 Simulated manipulations of a planar virtual linkage. Initial linkage position is shown in black. Linkage positions during manipulation are shown in grey. 78

5.6 Simulink diagram of the haptic manipulation of a planar virtual linkage. 79

5.7 Simulated planar manipulation of a three links virtual tool held from the COM of the distal link. 80

5.8 Simulated planar manipulation of a three links virtual tool held from the COM of the second link. 80

5.9 Testbed virtual environment for controlled linkage manipulation from any user-selected link. . . 82

5.10 Haptic manipulation from the distal link of the virtual linkage shown in Figure 5.9. Users apply a constant wrench $\mathbf{F}_h = (0.4\text{N} \quad 0\text{N} \quad 0\text{Nm})^T$. 82

5.11 Haptic manipulation from the second link of the virtual linkage shown in Figure 5.9. Users apply a constant wrench $\mathbf{F}_h = (0.4\text{N} \quad 0\text{N} \quad 0\text{Nm})^T$. 83

6.1 Realistic haptic manipulation of virtual tools using the local model of interaction proposed in this thesis. 85

6.2 Communication between the virtual environment (VE), the local model of interaction, and the haptic device. 86

6.3 Example haptic interaction during which the computational delay of the virtual environment results in delayed updating of the virtual tool contacts in the local model of interaction. 88

6.4 The influence of the geometry of the virtual environment on the interactions of the virtual tool. 88

6.5 Geometric information provided by typical simulation packages for each type of virtual tool contacts (Figures 6.5(a), 6.5(b), and 6.5(c)) and its representation in the local contact geometry (Figure 6.5(d)). 90

6.6 Partial virtual environment geometry available in the proposed local model of interaction. . . . 91

6.7 Local proxy deformation due to violation of new constraints. 92

6.8 Neighboring constraints included in the local model through the use of an ϵ-active virtual tool, obtained by sweeping a sphere of radius ϵ over the volume of the virtual tool. 93

6.9 The ϵ-active virtual tool eliminates the singularity in the constraint normal computation at a vertex-vertex contact between the virtual tool and the virtual environment by selecting a unique constraint normal direction. 93

6.10 One dimensional peg-in-hole task used to illustrate the influence of local geometry on user's perception of locally cluttered virtual environments. 94

6.11 Hand position for the simulated peg-in-hole task depicted in Figure 6.10(a). 95

6.12 Contact clustering in the virtual environment. 96

6.13 Typical commercial simulation packages use polygonal representations (in a planar virtual world) for both rounded objects and polygonal objects. 97

6.14 Experimental interaction used to investigate the passivity of the collision resolution approach proposed in this thesis. 101

6.15 Passivity of proxy dynamics in an impulse-augmented penalty local model of interaction. The user operates the virtual linkage in Figure 3.3(b) from its distal link. 102

6.16 Environment wrenches applied to users during user manipulation of the distal link of the articulated object in Figure 6.14(a) in an impulse-augmented penalty local model of interaction with perfectly plastic contacts, $e = 0$. 103

6.17 Environment wrenches applied to users during user manipulation of the distal link of the articulated object in Figure 6.14(a) in an impulse-augmented penalty local model of interaction with perfectly elastic contacts, $e = 1$. 104

6.18 Experiment used to investigate the haptic rendering of static contact. 105

6.19 Device trajectories obtained when the virtual tool shown in Figure 6.18(a) is pushed with a constant force towards the bottom-most horizontal wall of the virtual environment, as depicted in Figure 6.18(b). The intermediate representation ("IR"), the local model including active geometry ("LM_0"), and the local model including ϵ-active geometry within 5mm from the virtual tool ("LM_5") interface the haptic device to the virtual environment simulation. 105

6.20 Experiment used to demonstrate the role of ϵ-active geometry in improving user's perception of tight virtual clearances. 107

6.21 Device trajectories obtained when the virtual tool shown in Figure 6.20(a) is rotated by 90^o, inserted into the tight-fitting hole at the bottom of the virtual world, and shaken horizontally with a sinusoidally varying force, as depicted in Figure 6.20(b). The intermediate representation ("IR"), the local model including active geometry ("LM_0"), and the local model including ϵ-active geometry within 5mm from the virtual tool ("LM_5") interface the haptic device to the virtual environment simulation. 107

6.22 Device trajectories obtained when the virtual tool is inserted in the tight-fitting hole at the bottom of the virtual world shown in Figure 6.20(a) and shaken horizontally with a sinusoidally varying force, as depicted in Figure 6.20(b). The local model including active geometry ("LM_0") and the local model including predicted geometry within 5mm from the virtual tool ("LM_5") interface the haptic device to the virtual environment simulation. 109

6.23 Experiment used to investigate the realism of the haptic rendering of dynamic contact. 110

6.24 Device trajectories obtained when the virtual tool pushes the last link of the virtual linkage in Figure 6.23(a) with a constant force, as shown in Figure 6.23(b). The intermediate representation ("IR"), the local model including predicted geometry and static constraint ("LM_{SC}"), and the local model including prdicted geometry and moving (kinematic) constraints ("LM_{KC}") interface the haptic device to the virtual environment simulation. 110

D.1 The twin pantograph planar haptic interface. 134

D.2 The parameters of one pantograph linkage. The base joints are shown in black. The pantograph endpoint P_e is the joint between the two not grounded links. 135

D.3 The four channel architecture for 1DOF haptic interaction within virtual environments. Simulated blocks are shaded. 137

Glossary

β	damping coefficient
$\mathcal{J}_c \in \mathcal{R}^{d\times c}$	contact Jacobian
$\mathbf{\Lambda} \in \mathcal{R}^{6\times 6}$	operational space inertia matrix
$\mathbf{f} \in \mathcal{R}^3$	Cartesian space force
$\mathbf{F} \in \mathcal{R}^6$	Cartesian space wrench (force and torque)
$\mathbf{J}^{\#} \in \mathcal{R}^{c\times d}$	weighted generalized inverse of $\mathbf{J}$
$\mathbf{J}_h \in \mathcal{R}^{6\times d}$	Jacobian matrix of a linkage computed at the user's hand
$\mathbf{M}_Q \in \mathcal{R}^{d\times d}$, $\mathbf{M}_V \in \mathcal{R}^{6\times 6}$	metrics on the linear vector spaces Q, V
$\mathbf{n} \in \mathcal{R}^3$, $\mathbf{t} \in \mathcal{R}^3$	Cartesian space directions (of the contact, of the relative sliding velocity)
$\mathbf{n}_{tc} \in \mathcal{R}^d$	configuration space direction
$\mathbf{p} \in \mathcal{R}^c$	vector of magnitudes of frictionless collision impulses
$\mathbf{q} \in \mathcal{R}^d$, $\dot{\mathbf{q}} \in \mathcal{R}^d$, $\ddot{\mathbf{q}} \in \mathcal{R}^d$	configuration space position, velocity, acccleration
$\mathbf{R} \in \mathcal{R}^{3\times 3}$	rotation from world to proxy coordinates
$\mathbf{T} \in \mathcal{R}^d$	configuration space torque
$\mathbf{u} \in \mathcal{R}^6$	device control signal
$\mathbf{x}_h \in \mathcal{R}^6$, $\dot{\mathbf{x}}_h \in \mathcal{R}^6$, $\ddot{\mathbf{x}}_h \in \mathcal{R}^6$	body position, velocity, acceleration at the user's hand
$\boldsymbol{\omega} \in \mathcal{R}^3$	angular velocity of a rigid body
$\boldsymbol{\tau} \in \mathcal{R}^3$	Cartesian space torque
$\overline{\mathcal{J}}_c \in \mathcal{R}^{c\times d}$	dynamically consistent inverse of the contact Jacobian $\mathcal{J}_c$
μ, e	coefficient of friction, coefficient of restitution

$\underset{\sim}{\mathbf{c}} \in \mathcal{R}^3$, $\mathbf{v} \in \mathcal{R}^3$, $\mathbf{a} \in \mathcal{R}^3$	Cartesian position, velocity, acceleration of a point
Σ_i	i-th singular value of a matrix
$\widehat{\mathbf{D}}$, $\widehat{\mathbf{G}}$, $\widehat{\mathbf{n}}_c$, $\widehat{\mathbf{M}}_V$	value of $\mathbf{D}$, $\mathbf{G}$, $\mathbf{n}_c$, $\mathbf{M}_V$ in the virtual environment at the moment of the simulation update
a	stiction gradient
c	number of contacts of a contact group
C_b, C_u	rigid body constraints (bilateral and unilateral)
d	dimension of the configuration space of a contact group
F, T, Q, V	linear vector spaces
KE	kinetic energy
s	separation distance between two rigid bodies
Z	one dimensional impedance
z	strain of the bond between two contacting bodies
z_0	breakaway distance (maximum strain of the bond between two contacting bodies)
$\mathcal{R}(\mathbf{A})$, $\mathcal{N}(\mathbf{A})$	range space, null space of $\mathbf{A}$

Acknowledgements

I would like to acknowledge my thesis advisors, Professors Tim Salcudean and Elizabeth Croft, for their guidance and support throughout this work. They have allowed me the freedom to roam and have asked all the right questions. I feel very fortunate to have worked with them. If one day somebody will be as grateful for having worked with me as I am today for having worked with Tim and Elizabeth, then I have learned something from them and their efforts have not been wasted.

I would also like to thank my colleagues in the Robotics and Control Laboratory for the times we shared and for the friendships we built. I hope our friendship will endure through years to come. My special thanks go to Shahin, for his patience to discuss system dynamics with me. I am also greatful to my colleagues in the Industrial Automation Laboratory for our instructive and enjoyable laboratory meetings.

Above all, I thank my family for supporting all I do. Everything would be worthless without you.

Chapter 1

Introduction

This thesis is concerned with providing realistic force, or haptic, feedback to users interacting with rigid multibody virtual environments. Lifelike forces give users a sense of “presence” in the computer-generated virtual environment; that is, they make users feel as if they are directly manipulating the rigid virtual objects. Realistic force interactions are important for physical skill training and for gaining an understanding of the virtual world that cannot be obtained using other human-computer communication paradigms. For example, astronauts may practice the operation of specialized tools under conditions similar to those during the space mission before the tools are manufactured; engineers may recognize limitations of their designs by manipulating virtual prototypes; students may become familiar with various physical phenomena by interacting with virtual objects. Despite significant potential benefits, commercial applications enabling force interactions with rigid multibody virtual worlds do not exist yet. Such interactions continue to be a focus of on-going research, including of this thesis.

It is challenging to apply lifelike forces to users manipulating rigid virtual objects because the key components of a haptic interaction system impose conflicting requirements on the interaction (see Figure 1.1). Users require virtual forces at rates of the order of 500Hz or higher during kinesthetic interaction with rigid objects [28, 34, 135] so that they do not perceive artifacts such as vibrations. Additionally, they exchange energy with the simulation. The device controller needs similar high “haptic rates” in order to create an adequate perception of rigidity. It also requires fixed force update rates and passive virtual environments to stably transfer energy between the user and the simulation [3]. Moreover, the device can display only limited stiffnesses to the user [40]. However, typical interactive physically-based simulations of rigid multibody virtual environments either have rendering frequencies only of the order of tens of Hz [140, 149], and can slow down further during complex interactions, or have geometry-dependent stiffness [42] that cannot be guaranteed for arbitrary interactions [59]. In other words, while the device requires realistic forces updated at guaranteed high rates and arising from a limited environment stiffness, existing simulations provide these forces with uneven computational delay or cannot guarantee the stiffness that they represent. In prior haptics research, three approaches have been developed to address the de-stabilizing effect of the simulation.

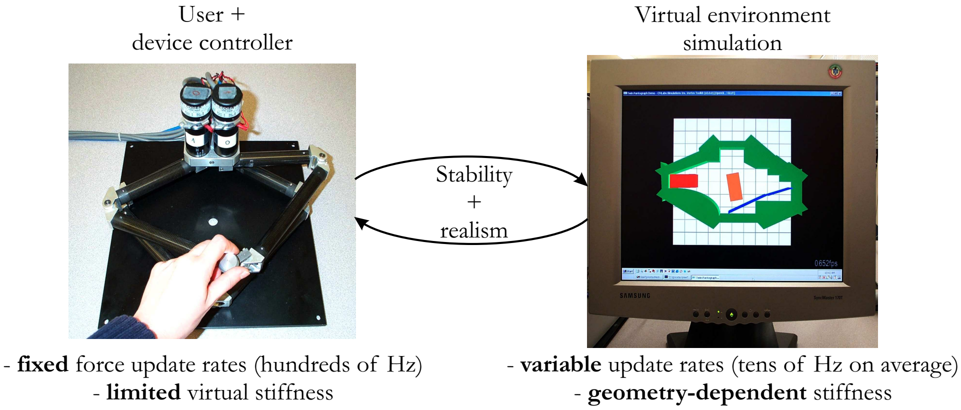

User +
device controller
Virtual environment
simulation
Stability
+
realism
- **fixed** force update rates (hundreds of Hz)
- **limited** virtual stiffness
- **variable** update rates (tens of Hz on average)
- **geometry-dependent** stiffness

Figure 1.1: The key components of a haptic interaction system and their conflicting requirements for stability and realism.

A first approach used control techniques to guarantee stable user interaction within virtual environments with arbitrary stiffness and/or computational delay (Figure 1.2). The most common method separates the simulation from the device through a "virtual coupler" [42]. The virtual coupler is a controller that limits the stiffness displayed to the user, eliminating the potentially destabilizing effect of the geometry-dependent environment stiffness. To date, it has been used for rendering both point [128, 140] and rigid body [22] interaction within rigid multibody virtual environments. The virtual coupler guarantees stable manipulation of passive nondelayed virtual worlds [3]. However, the restriction to passive nondelayed environments is quite limiting, since existing nondelayed rigid multibody simulations use penalty-based dynamic algorithms and fixed step forward integrators that are nonpassive [31]. Furthermore, the virtual coupler introduces perceptual artifacts (such as contacts that pull on the user) and does not allow users to perceive physical phenomena that rely on fast force transitions (such as collisions and stick-slip friction). Recently, a control technique was designed in [65] that monitors the energy produced by one or more simulation blocks and dissipates this energy when necessary. The technique guarantees stable user interaction within any virtual environment that produces less energy than the device can dissipate, regardless of the computational delay of the simulation. However, the energy dissipation mechanism introduces perceptual artifacts (e.g., it allows contacts to pull on the user) that compromise the realism of the interaction.

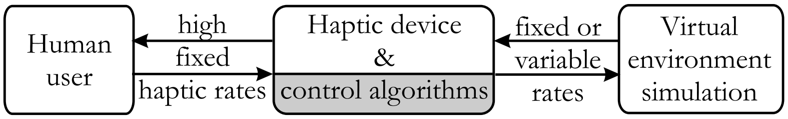


Figure 1.2: Alleviation of the effect of the computational delay and the geometry-dependent environment stiffness via control.

A second approach improves the real time performance of the simulation via new collision detection methods that eliminate the collision detection bottleneck and significantly decrease the computational delay of the virtual environment. The new methods take advantage of hierarchical spatial bounding volumes [59, 75, 85], incremental distance computations [76, 83], and multiresolution bounding hierarchies [117] to detect contacts and to approximate penetration distances at the high haptic speeds. Combined with penalty-based dynamic algorithms and fixed step forward integration routines that require little additional computation time, the new collision detection techniques allow the simulation to run synchronously with the haptic controller (see Figure 1.3). However, instability may arise if the user exerts too much force which causes a large amount of penetration [59], or in non-convex contact configurations [85].

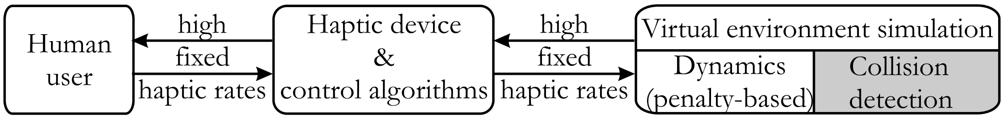


Figure 1.3: Elimination of the computational delay of the simulation via new collision detection algorithms.

A third approach alleviates the haptic rate demand on the virtual environment by decoupling the force control loop from the simulation (see Figure 1.4). Therefore, virtual environments of increased complexity, based on passive algorithms, can be rendered. In this approach, users feel realistic forces only if the setpoints of the control loop over one simulation step suitably approximate the forces acting on the virtual object manipulated by the user (the virtual tool). Hence, a local model of the interaction (i.e., a reduced simulation that runs at the fixed haptic rate) must be available in the force control loop. A first local model for point interaction with rigid virtual objects was proposed in [2]. It incorporates the geometry of the constraint restricting the user's motion. This local model shifts the computational delay of the simulation from delay in computing the interaction forces to delay in updating the local geometry. The model was refined in [100, 149] to ensure geometry continuity at the model updates. However, its extension to haptic interaction with a rigid multibody virtual world through a virtual tool led to instability during tightly constrained motions of the virtual tool, such as during peg-in-hole insertion [21]. Therefore, the local geometry was used to constrain only the translation of the device and a virtual coupler was used to constrain its rotation.

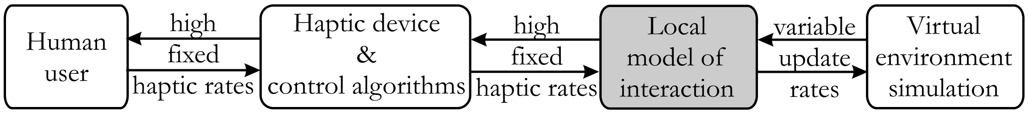

Human user
high fixed haptic rates
Haptic device & control algorithms
high fixed haptic rates
Local model of interaction
variable update rates
Virtual environment simulation

Figure 1.4: Mitigation of the effect of the computational delay of the simulation via a local model of interaction.

None of the approaches described above allows realistic and guaranteed stable haptic manipulation of rigid virtual objects. Such interaction continues to be a challenging proposition. Both new control and new simulation methods need to be developed in order to increase the physical accuracy of the haptic rigid object manipulation while maintaining its stability.

1.1 Objective

The goal of this thesis is to increase the realism of the force interaction within rigid multibody virtual environments through modeling and simulation methods that enable efficient and transparent haptic rendering of rigid body motion with constraints. In particular, general-purpose simulation methods are proposed that model interactions convincingly, yet suitably for being applied to the user's hand by the device controller. Such simulation methods have not been available in prior research. A further objective of this work is to develop local approximations of the proposed simulation approaches that adequately represent the interaction and run at the high haptic speeds regardless of the complexity of the virtual environment.

1.2 Contributions

This work contributes to: (i) physically-based modeling of rigid multibody virtual environments; (ii) haptic manipulation of virtual linkages; and (iii) local models of interaction for rigid multibody virtual worlds.

- **Physically-based modeling of rigid multibody virtual environments**

  Collisions are ubiquitous during physical manipulations of rigid objects. In haptics, collisions felt by the user enhance the perceived rigidity of the virtual contacts [90, 125]. To increase the realism of the haptic interaction within rigid multibody virtual environments, this thesis develops a simulation approach that enables users to perceive collisions. The approach uses a contact model compatible with fixed step integrators and closed-form dynamics algorithms. Specifically, contacts are infinitely stiff upon contact and have limited stiffness during contact (see Figure 1.5). Furthermore, a new simultaneous collision resolution algorithm is employed upon contact. The algorithm is proven to never increase the kinetic energy of the virtual environment. Users feel collisions through impulsive forces. They feel contacts through penalty and friction forces.

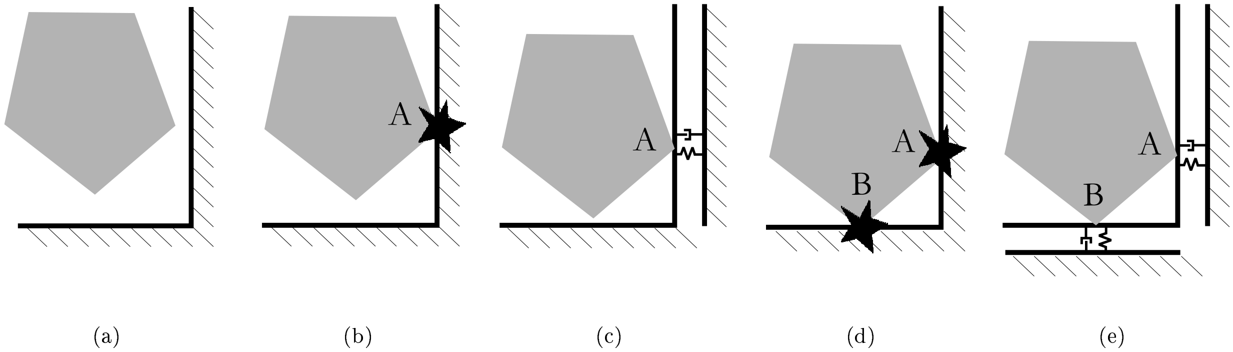


Figure 1.5: The contact model the enables users to feel collisions: contacts have infinite stiffness upon contact and finite stiffness during contact. Users feel impulsive forces upon contact and penalty and friction forces during contact.

  Although developed for haptics, the simulation approach proposed in this thesis can be used in any application that requires rigid multibody virtual environments with interactive performance. The approach provides a compromise between the efficiency of penalty-based simulations [62, 111] and the accuracy of constraint-based simulations [7, 10, 13, 35, 63, 66, 70, 110, 127, 141, 143].

- **Haptic manipulation of virtual linkages**

  In contrast to prior work that allows operation of spatial (planar) virtual linkages from links with at least 6 DOFs (3 DOFs) [114, 128, 140], this thesis develops methods that enable realistic rigid body manipulation of linkages from any user-selected point. In particular, users are enabled to manipulate links with motion limited by the linkage topology and to convincingly perceive motion restrictions due to other virtual objects and to the linkage itself. This is important in applications like training for laparoscopic procedures, for example, when the scope restricts the motion of the virtual tool at the point of entry. Motion constraints due to other virtual objects are imposed through contact (impulsive, penalty, and friction) forces transformed to the user's hand in a coordinate-invariant manner. The linkage topology is imposed through penalizing users when they violate the configuration manifold of the linkage (see Figure 1.6).

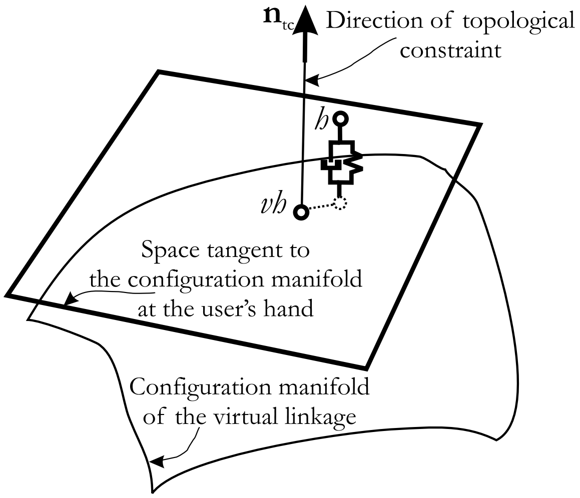

$\mathbf{n}_{tc}$
Direction of topological constraint
h
vh
Space tangent to the configuration manifold at the user's hand
Configuration manifold of the virtual linkage

Figure 1.6: Users perceive topological constraints through penalties applied to the device when their hand leaves the configuration manifold of the linkage (vh is the position and orientation of the user's hand on the virtual linkage, while h is their real position and orientation).

- **Local models of interaction within rigid multibody virtual worlds**

Regardless of the efficiency of the algorithms employed, the computational performance of rigid multibody virtual environments depends at least linearly on the complexity of the virtual scene to be simulated (i.e., its contact geometry [12,59,76,83,84] and the dimension of its configuration space [48,91,127]). To enable haptic interaction within virtual worlds of arbitrary complexity, this thesis develops a local model of rigid body interaction that is simple enough to run at the haptic rates and general enough to interface a device to a virtual environment regardless of the algorithms used to simulate the virtual world. This model is the first local model that includes a dynamic proxy of the virtual tool manipulated by the user and active and predicted motion constraints imposed on the virtual tool by nearby objects (see Figure 1.7). Geometry prediction enables realistic manipulations of the virtual tool through tight clearances. The dynamic proxy encapsulates motion constraints imposed on users by articulated structures, allowing them to manipulate linkages. Furthemore, the haptic controller can coordinate both forces and motions between the haptic device and the proxy. Coordination of forces increases transparency and enables users to feel collisions, thus enhancing the perceived local contact rigidity. Position coordination allows stiffer objects to be manipulated.

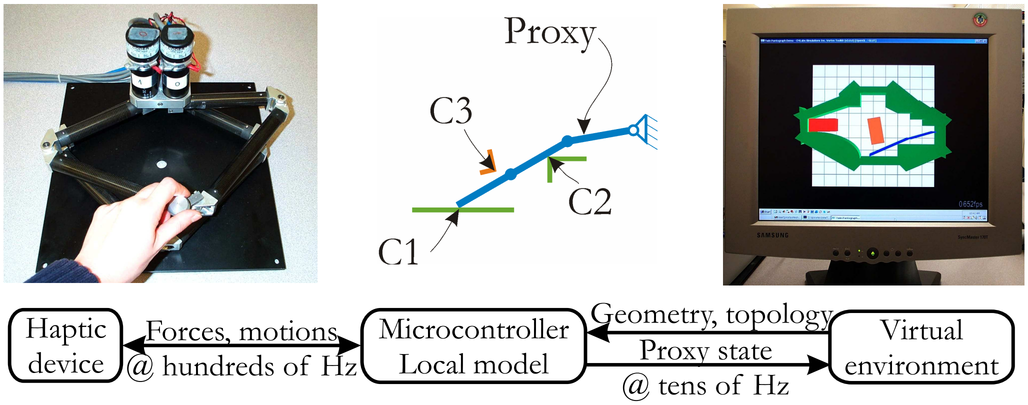


Figure 1.7: The local model of rigid body interaction includes a dynamic proxy of the virtual tool and active and predicted motion constraints imposed on the virtual tool by nearby objects.

1.3 Organization of the thesis

The remainder of the thesis is arranged as follows:

Chapter 2: The state of the art in physically-based rigid multibody simulation and impact modeling is presented, focusing on methods with interactive performance. Simulation and control techniques employed for haptic interaction within rigid multibody virtual environments are also overviewed.

Chapter 3: This chapter describes the planar haptic interaction system used to experimentally validate the simulation and the control methods proposed in the present thesis.

Chapter 4: A new approach to haptic rendering of rigid body contact is introduced that enables users to feel collisions and, thus, enhances the perceived rigidity of the virtual objects. The chapter describes the underlying contact model, as well as the simulation of impulsive and contact interactions. A new, passive, multiple collision resolution technique is proposed and modified to account for the practical limitations of haptic devices. Simulations and controlled experiments are used to validate the proposed rigid contact rendering approach. A friction model developed in computational dynamics is, for the first time, used to haptically render dry friction. The efficacy and the advantages of this model are evaluated by comparing it to models priorly used in haptics through simulations and experiments.

Chapter 5: In this chapter, simulation and control methods are devised that enable realistic manipulation of virtual linkages. Linkage contacts are modeled as described in Chapter 4. They are applied to the user's hand using a coordinate-invariant representation. The inertia and the topology of the virtual linkage are rendered to users through device control. The limitations of such control are overcome through a physically-motivated technique that convincingly restricts the motion of the device as required by the topology of the virtual linkage. Realistic and unrestricted operation of virtual linkages is demonstrated through simulated and experimental manipulations of a planar virtual linkage from various links.

Chapter 6: A local model of rigid body interaction is developed and used for haptic manipulation of rigid virtual objects. The communication between the local model and the device, and the local model and the virtual environment is described. Geometry and dynamics techniques ensuring model continuity at simulation updates are detailed. A numerically efficient implementation of the methods in Chapters 4 and 5 is proposed. The novel local model of rigid body interaction is validated experimentally by using it to interface a planar haptic device to a virtual environment generated using a commercial simulation package.

Chapter 7: This chapter summarizes the contributions and limitations of the work described in this thesis and discusses directions for future work.

Chapter 2

Physically-based rigid multibody simulations

In haptic applications, both the computational speed and the physical accuracy of the simulation are essential for the user's sense of presence in the virtual environment. Physically-based rigid multibody simulations achieve interactive performance by balancing efficiency and accuracy in various ways. This chapter starts by surveying interactive simulations and the modeling methods they employ. The survey emphasizes the potential of existing techniques to achieve the requirements of haptic interaction, namely physical realism and guaranteed real time performance. The chapter ends by overviewing haptic manipulation of rigid multibody virtual environments, including simulation and interaction control techniques, local models of interaction, and impact and friction rendering.

2.1 Interactive simulations

Simulation algorithms implement a loop consisting of three main steps (see Figure 2.1):

- collision detection: the body contacts are computed;
- dynamic response: the equations of motion (EOMs) are solved in accordance with the new set of contacts;
- time integration of the system state: the body positions and velocities at the next time step are computed.

All three steps contribute both to the physical accuracy and to the efficiency of the simulation. For example, recent collision detection algorithms [75, 76, 83, 85, 108] have significantly alleviated the collision detection bottleneck and correspondingly improved the simulation performance. However, only work addressing the dynamic response is relevant to the scope of this thesis and, therefore, surveyed in this section. The focus is on the ability of existing dynamic response techniques to realistically represent rigid multibody interaction while achieving the real-time performance required by the haptic controller.

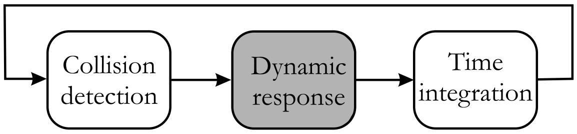


Figure 2.1: The simulation loop and the scope of the survey: prior work in dynamic response.

Dynamic response includes the formulation of the EOMs of rigid multibody systems starting from the fundamental principles of the Newtonian mechanics (principle of virtual displacements, Gauss' least action principle, etc.), and their solution. The EOMs of a multibody system are derived as a set of ordinary differential equations (ODEs) using either positions and velocities or positions and momenta as state variables. However, general purpose simulations use positions and velocities as state variables exclusively. In their most general form, the EOMs of a system of b rigid bodies with d DOFs with positions and velocities as state variables are a set of $n \geq d$ second order ODEs [146]:

$$\mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{Q}^c(\dot{\mathbf{q}},\mathbf{q},t) = \mathbf{Q}(\dot{\mathbf{q}},\mathbf{q},t) \quad \Rightarrow \quad \mathbf{M}(\mathbf{q})\,\ddot{\mathbf{q}} + \mathbf{Q}^c_i(\dot{\mathbf{q}},\mathbf{q},t) + \mathbf{Q}^c_{ni}(\dot{\mathbf{q}},\mathbf{q},t) = \mathbf{Q}(\dot{\mathbf{q}},\mathbf{q},t)\,. \tag{2.1}$$

In Equation (2.1), $\mathbf{M} \in \mathcal{R}^{n\times n}$ is the system mass matrix; $\ddot{\mathbf{q}} \in \mathcal{R}^n$, $\dot{\mathbf{q}} \in \mathcal{R}^n$, $\mathbf{q} \in \mathcal{R}^n$ are the system acceleration, velocity, and position vectors, respectively; $\mathbf{Q} \in \mathcal{R}^n$ is a vector that includes external forces as well as Coriolis, centrifugal, and gravitational effects; $\mathbf{Q}^c = \mathbf{Q}^c_i + \mathbf{Q}^c_{ni} \in \mathcal{R}^n$ is the vector of constraint forces, whose components are $\mathbf{Q}^c_i \in \mathcal{R}^n$, the vector of ideal constraint forces (they do no work and are along the constraint directions), and $\mathbf{Q}^c_{ni} \in \mathcal{R}^n$, the vector of nonideal constraint forces (they do work and are in the constraint plane). Note that Equation (2.1) does not represent the EOMs as they are directly derived from the principles of mechanics; rather, it represents the EOMs rearranged for computational purposes. Unknown forces and accelerations are grouped on the left hand side of this equation, while known forces are on its right hand side. If the b bodies have c contacts, the n-dimensional Equation (2.1) has $n + 2c$ unknowns: the system acceleration $\ddot{\mathbf{q}}$, and the constraint forces $\mathbf{Q}^c_i$ and $\mathbf{Q}^c_{ni}$. Additional equations are needed to find the dynamic response of the system, i.e., to compute the system acceleration $\ddot{\mathbf{q}}$ to pass it to the time integration routine.

The additional equations are given by the constraints and by the laws of dry friction. The constraints enforce the rigidity assumption by requiring the bodies not to overlap. Bilateral constraints represent permanent connections between bodies, or *joints*, whereas unilateral constraints represent temporary connections, or *contacts*. If the system has j joints and c contacts, then there are j bilateral constraints:

$$C_{b_i}\left(\dot{\mathbf{q}}, \mathbf{q}, t\right) = 0 \quad \forall i = 1, \cdots, j, \tag{2.2}$$

and c unilateral ones:

$$C_{u_i}\left(\mathbf{q}, t\right) \geq 0 \quad \forall i = 1, \cdots, c. \tag{2.3}$$

Additionally, the Coulomb law of friction requires the nonideal and the ideal constraint forces to be related through:

$$||Q^c_{ni_i}|| \leq \mu_i ||Q^c_{i_i}|| \quad \forall i = 1, \cdots, j + c, \tag{2.4}$$

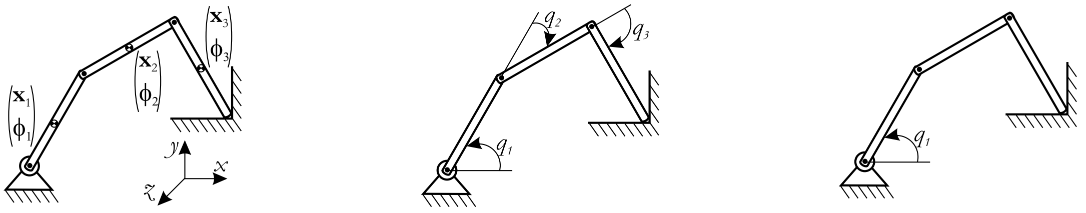
where the index i selects the i-th vector component and μ_i is the coefficient of dry friction at the i-th contact. The constraints in Equation (2.2) and the EOMs form a set of second order differential algebraic equations (DAEs). Integration of DAEs is unsuited for real-time performance. Therefore, in interactive simulations, these DAEs are transformed into ODEs by appending the constraints to the dynamics at the acceleration level (i.e., by differentiating Equation (2.2) twice with respect to time):

$$\mathbf{A}_{j\times n}\left(\dot{\mathbf{q}}, \mathbf{q}, t\right)\ddot{\mathbf{q}} = \mathbf{b}_{j\times 1}\left(\dot{\mathbf{q}}, \mathbf{q}, t\right). \tag{2.5}$$

In Equation (2.5), $\mathbf{A}$ is the Jacobian of the constraints (the constraint matrix) and $\mathbf{b}$ groups all other terms obtained by differentiation. Dynamic response requires the systems acceleration $\ddot{\mathbf{q}}$ to be computed. However, Equations (2.1) and (2.5) form a system of $n + j$ equations with $n + 2j + 2c$ unknowns, $\mathbf{Q}^c_i$, and $\mathbf{Q}^c_{ni}$. The additional equations necessary for computing the dynamic response depend on the employed model of rigid body contact, as described in Section 2.1.2. The numerical efficiency and the physical accuracy of the dynamic response depend both on the coordinates used to describe the system state and on the model of rigid body contact, as described in Sections 2.1.1 and 2.1.2.

2.1.1 Coordinates for rigid multibody dynamics

Rigid multibody simulations describe the system state using Cartesian, generalized (dependent), or independent coordinates (see Figure 2.2). The coordinate system used depends on whether bilateral constraints are distingushed from unilateral ones.


$\begin{pmatrix}\mathbf{x}_1\\ \phi_1\end{pmatrix}$
$\begin{pmatrix}\mathbf{x}_2\\ \phi_2\end{pmatrix}$
$\begin{pmatrix}\mathbf{x}_3\\ \phi_3\end{pmatrix}$
y
x
z
q_2
q_3
q_1
q_1

(a) Cartesian coordinates.

(b) Generalized coordinates.

(c) Independent coordinates.

Figure 2.2: Coordinates used to describe the system state in rigid multibody simulations.

Cartesian coordinates (Figure 2.2(a)) do not differentiate bilateral constraints from unilateral ones. They incorporate no topological information into the EOMs. The dynamic model of each body is independent of all others and constraint equations are added or deleted from the constraint matrix $\mathbf{A}$ as needed [56]. Cartesian coordinates are best suited for general purpose simulators. They are used in many implementations, developed in computational dynamics [56, 57, 66, 116, 153, 154]), in computer graphics [5, 12, 16, 32, 37, 62, 104, 109, 110, 122, 134], and in haptics [34, 85, 117]. However, models in Cartesian coordinates result in the largest number of dynamic equations (three per body for planar interaction and six for spatial interaction) and are prone to constraint violation. The numerical drift is addressed by constraint stabilization [5, 19, 37] and by projection methods [82, 98], which integrate only d independent coordinates and use the constraint equations to compute the remaining $n - d$ position variables.

Generalized coordinates (Figure 2.2(b)), e.g., relative coordinates [48] and natural coordinates [51], incorporate the bilateral (joint) constraints into the EOMs and explicitly represent only the unilateral (contact) constraints. Hence, the number of EOMs to be solved is smaller, equal to the number of DOFs of the system. Moreover, the system evolves on its configuration manifold and the bilateral constraints are automatically satisfied. Efficient serial algorithms are proposed in [25, 48, 73, 74, 91, 107] for loop-free topologies and in [72] for topologies with internal loops, while efficient parallel algorithms for linkages with or without loops are presented in [49, 50]. Collision resolution and contact force determination methods for tree-like systems are described in [127], where the increased numerical efficiency of the coordinate representation allows the simulation of a 26 DOFs tree-like robot at the haptic rates. The significant computational advantage of generalized coordinates has led to the development of several simulators with interactive performance [13, 51, 128, 140, 141], some of which have also been used for haptic interaction [128, 140].

Independent coordinates (Figure 2.2(c)) are the minimum number of coordinates required for describing a multibody system, i.e. $n = d$. Minimal representations result in a minimum number of EOMs and no numerical drift, since all constraints are incorporated in the choice of independent coordinates. However, the set of independent coordinates changes as the constraint configuration changes. This set cannot be derived numerically at present. Hence, in interactive applications, the efficiency gained by embedding the constraints in the EOMs may be offset by the symbolic calculations needed to derive a new system model at configuration changes. An application-specific simulator using independent coordinates has been demonstrated in [52, 54] for haptic rendering of piano playing. In this simulator, all constraint configurations are known prior to run time. Hence, dynamic models corresponding to these configurations are derived off-line and then combined on-line using a finite state machine. Recently, the design of a general purpose haptic simulator using independent coordinates has been outlined in [53]. This design proposes to achieve interactivity by decoupling the symbolic computation of the EOMs from their numerical solution through running the symbolic model formulation in a thread outside the main simulation loop. Further work is required to demonstrate the interactive performance of this design.

In addition to the choice of coordinates, the enforcement of the rigidity and of the dry friction assumptions also influence the computational efficiency and the physical accuracy of rigid multibody simulations. This influence is discussed in the following section by overviewing dynamic models of rigid multibody systems available in prior work.

2.1.2 Dynamic models of rigid multibody systems

Rigid body dynamics have been a topic of active investigation for more than three hundred years. This long-lived interest has arisen from the mathematical difficulties accompanying two widely used modeling simplifications: the rigid body and the Coulomb dry friction assumptions. These simplifications bring about issues related: (i) to the existence and the uniqueness of solutions of the EOMs of rigid multibody systems with unilateral constraints and dry friction [7,47,92,94,102,118]; and (ii) to the algorithms approximating these solutions [6,7,143,144].

Physically-based rigid multibody simulations differ in how they enforce these simplifications and, therefore, in how they use the dynamics given by Equation (2.1). Under the rigid body assumption, Newtonian mechanics distinguishes three contact states: free motion, collision, and contact. Under the Coulomb friction assumption, contact can further be divided into sliding and sticking contact. Collisions (or impacts) are phenomena with much faster time constants then contacts. They are often modeled as instantaneous events which result in velocity discontinuities. Contacts are modeled as finite duration events which result in acceleration discontinuities. Based on the contact states they model, rigid multibody simulation methods can be classified in: (i) penalty-based; (ii) constraint-based; and (iii) impulse-based approaches.

2.1.2.1 Penalty-based simulations

Penalty-based simulations model only free motion and contact (Figure 2.3). They enforce rigidity only approximately, by introducing restoring forces when objects interpenetrate. Hence, they incorporate the penalized constraints into the EOMs. Since they only loosely enforce constraints, penalty methods sacrifice accuracy for robustness (singular configurations pose no problem) and a reduced set of equations (reaction forces are eliminated from Equation (2.1), as they are computed from local contact geometric information). Thus, penalty methods are compatible with the fixed time step integrators required in haptics because they do not need to exactly determine when new contacts arise.

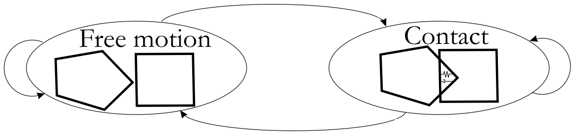


Figure 2.3: The contact states in a penalty-based rigid multibody simulation.

A main limitation of penalty simulations is that large penalties are necessary for acceptable accuracy. This induces numerical ill-conditioning during integration and requires the penalties to be adjusted for different simulation conditions. However, highly realistic computer animations of rigid bodies were demonstrated in [109, 121] and a general purpose simulation package was developed in [57,58].

A more important limitation for haptics is the fact that the environment impedance depends on the contact geometry and cannot be bounded for interactions within arbitrary virtual worlds (i.e., for arbitrary contact geometries). This is illustrated by the example in Figure 2.4. In this example, the contact between the rectangular object and the environment is represented through $c = 8$ point contacts and the environment stiffness is represented as:

$$\mathbf{K} = \begin{pmatrix} 0 & c \cdot k_{contact} & 0 \end{pmatrix}^T, \tag{2.6}$$

where $k_{contact}$ is the contact stiffness. Regardless of the chosen contact stiffness, contact geometries may exist that induce instability. Instability may arise because the environment stiffness is outside the region of conditional stability corresponding to the given time step of the forward integration routine used in the haptic simulation. As discussed in [40], instability may also arise due to an environment stiffness larger than the Z-width of the device (the impedance range that the device can render stably) [29].

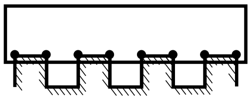


Figure 2.4: A rectangular object interacting with a virtual wall. Each surface contact is represented through two point contacts.

Despite the difficulties ensuing from the conditional stability of penalty methods, their low computational requirements and ease of implementation are very attractive for haptics. Several implementations of such methods exist [59,75,76,83,85,103,117].

2.1.2.2 Constraint-based simulations

Constraint-based simulations model all contact states and enforce exact rigidity by computing either collision impulses or contact forces to prevent body interpenetration (Figure 2.5). Interactive constraint-based simulations are more difficult to implement because they need to detect contact state changes at run time. This means that, given the set of geometric contacts reported by the collision detection algorithm, constraint-based simulations must determine the set of "active" contacts, i.e., the set of contacts at which objects interact.

Such detection is possible if the system dynamics is derived based on the complementarity rule (Figure 2.6), also known as the corner law or Signorini's conditions [120]. In the direction normal to the contact, the rule

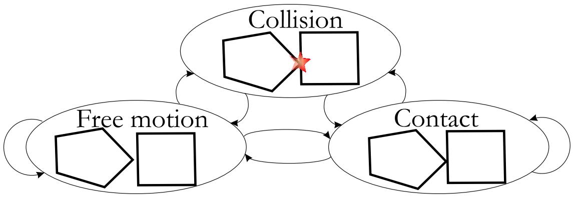
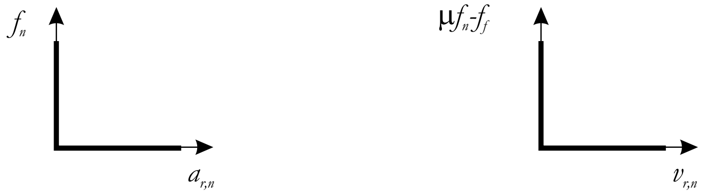


Figure 2.5: The contact states in a constraint-based rigid multibody simulation.

states that either the relative acceleration $a_{r,n}$ is zero and the contact force f_n is nonnegative[1], or the relative acceleration is nonnegative and the contact force is zero. In the contact plane, it states that either the relative tangential velocity $v_{r,n}$ is zero and the friction force f_f is in the friction cone (i.e., $\mu f_n - f_f \geq 0$, where μ is the coefficient of dry friction), or the relative tangential velocity is nonnegative and the friction force is on the boundary of the friction cone (i.e., $\mu f_n - f_f = 0$).



(a) The normal contact direction.

(b) The contact plane.

Figure 2.6: The corner law.

Two types of complementarity-based dynamics formulations exist: acceleration-force methods [93,94] and velocity-impulse methods [112]. Correspondingly, event-based and time-stepping rigid multibody simulations were developed [27]. Event-based simulations [11,120,122,127] integrate the system dynamics between collision events, resolve occurring collisions, and reset the integrator before continuing to advance the system state in time. Such algorithms are not directly useful in haptics because they require variable step size integrators; i.e., there is no guaranteed completion time.

Time-stepping simulations [6–8, 37, 104, 142, 143] solve a time discretization of the system dynamics that includes the impact rules and the perfectly rigid constraints enforced at the velocity level. These algorithms are compatible with the fixed step haptic integrators, but have no guaranteed completion time. This is because

[1] The contact force is positive if it pushes the bodies apart. The relative acceleration and the relative velocity are positive if they are such that the bodies separate.

they can avoid constraint drift only through accurate collision detection and the reformulation of the system dynamics at each contact configuration. Since there exists no *a priori* bound on the number of collisions that can occur during one simulation step, the computation time is unpredictable. Time-stepping algorithms with guaranteed completion time include suitable constraint stabilization techniques [6, 37]. In [37], constraint stabilization is formulated as a complementarity problem solved at the end of the simulation step. Besides the computational price paid by solving two linear complementarity problems (LCP) per step, the method is not shown to ensure the passivity of the simulated system. In [6], constraint stabilization is achieved by augmenting the complementarity-based system dynamics with a term that depends on the geometrical constraint infeasibility. Hence, no additional computational costs are incurred. However, the method is proven to ensure the passivity of the simulated system only for perfectly plastic collisions between smooth convex objects. Further work is needed to extend the method to elastic and partially plastic collisions between polyhedral objects, which are only locally smooth and may not be convex.

2.1.2.3 Impulse-based simulations

Impulse-based simulations model only free motion and collisions (Figure 2.7). They enforce rigidity exactly, by representing any interaction as staggered collisions and computing impulses that prevent body interpenetration. Contact is represented as a series of microcollisions. Friction is dealt with during microcollisions. The impulse-based technique was proposed in [62] and was developed into a general purpose simulation package in [110]. Recently, the physical accuracy of the method was traded off for efficiency by adding heuristics that enable interactive performance for simulations as large as 1000 cubes in an “hourglass” [134]. These heuristics are visually acceptable, since the number of virtual objects is too large for humans to be able to predict the natural motion of individual bodies. Impulse-based simulations are suitable for simulating systems with frequently occurring impacts (such as a part feeder, for example), but inefficient at simulating systems with joints and persisting contacts.

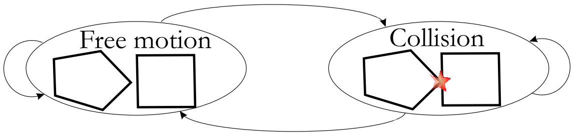


Figure 2.7: The contact states in an impulse-based rigid multibody simulation.

A haptic implementation of the impulse-based method was proposed in [34], but the results were not encouraging. In particular, the perception of contact was not convincing and users could not feel friction, regardless of the fact that it was built into the collision model [34]. There are two possible causes for this. First, the representation of contact through series of microcollisions could be satisfactory for visual feedback, but inappropriate for kinesthetic feedback. Second, it might be challenging to reconcile fixed step integration with exact enforcement of rigidity in a physically meaningful manner. For example, invalid impact states can result during a fixed step impulse-based simulation. These are states where two objects are in contact, but they are separating from each other and collision impulses cannot be computed. Because only collisions are modeled, contact can be maintained only by assigning a "computational" contact state to the objects. Assignments of a "computational" contact state other than those used in [34] might be needed to enable users to feel friction.

This overview of physically-based simulation techniques with interactive performance reveals that no type of coordinates or simulation method is ideally suited for generating the rigid multibody virtual environment in haptic applications. For example, further work is required for guaranteed stable interaction with penalty-based simulations and constraint-based rigid multibody haptic simulations have yet to be demonstrated. Additional difficulties in rendering rigid contact arise from the fact that the haptic controller has limited capabilities (limited bandwidth and motor power, for example). Therefore, it cannot prevent user penetration into the constraints. Nevertheless, psychophysical experiments [90, 125] have demonstrated improved perception of contact rigidity if large forces are applied to users upon contact. The present research associates such large forces with collisions and develops a collision resolution method suitable for haptic interaction. To put this method in perspective, relevant work in impact simulation is overviewed in the following section. As before, the emphasis is on the ability of prior models to realistically model collisions while achieving real-time performance.

2.2 Impact simulation

Impact simulation is important in a large number of applications, including analysis of vibrations in machines [120], simulation of robotic pushing [101] and orienting [55] operations, virtual prototyping [1], and interactive computer graphics [134]. At the same time, it is a challenging endeavor, mostly because the many physical phenomena occuring during collisions (restitution, tangential compliance, jamb, vibrations) are difficult to capture into simple models that apply to both single and multiple collisions. Therefore, many collision (or impact or restitution) laws (or rules) were proposed that aim to realistically predict the collision outcome. They fall into three categories: compliant, differential, and algebraic collision laws [27].

Compliant [57, 86, 87, 106] and differential [15, 18, 23, 24, 78, 99, 115, 145] impact laws treat collisions as phenomena with a finite duration, derive the corresponding equations of motion (under the assumption of constant body positions), and integrate them in order to compute post-impact body velocities. Such rules are adequate for resolving simultaneous collisions with friction [86, 99], i.e., they predict collision outcomes that agree with basic constraints like nonnegative energy dissipation and the Coulomb dry friction inequality at the impulse level. However, compliant and differential restitution laws require much smaller integration steps during collisions. Hence, they are not suitable for simulations with interactive performance.

Algebraic collision laws [4, 7, 26, 35, 63, 120, 137, 142, 143, 151] ignore the mechanics of the interaction and predict the collision outcome using various generalizations of Newton's and Poisson's rules. Newton's law relates the pre-collision ($v_{r,n0}$) and post-collision ($v_{r,n}$) relative normal contact velocities[2] through the coefficient of restitution e:

$$v_{r,n} = -e v_{r,n0}. \tag{2.7}$$

Poisson's rule decomposes the collision into compression and restitution and relates the normal impulses during restitution $p_{n,r}$ and compression $p_{n,c}$ through the coefficient of restitution e:

$$p_{n,r} = -e p_{n,c}. \tag{2.8}$$

The coefficient of restitution $e \in [0, 1]$ describes the nature of the collision, with $e = 1$ corresponding to perfectly elastic collisions (zero energy loss), $e = 0$ corresponding to perfectly plastic collisions (maximum energy loss), and intermediate values corresponding to plastic collisions (some energy loss). Friction is incorporated into collisions through Coulomb's dry friction law, by requiring that:

$$p_t \leq \mu p_n, \tag{2.9}$$

[2] In this work, "normal" indicates the component of a vector along the direction normal to the contact plane, while "tangential" indicates the component of the vector in the contact plane. The normal direction and the tangent plane are well defined almost everywhere.

if Newton's restitution hypothesis is employed, or:

$$\begin{array}{rcl} p_{t,r} & \leq & \mu p_{n,r} \\ p_{t,c} & \leq & \mu p_{n,c} \end{array}, \tag{2.10}$$

if Poisson's rule describes the collision. In Equations (2.9) and (2.10), μ is the coefficient of friction, and the subscripts n and t indicate vector components along the contact normal and in the contact (tangent) plane, respectively. Multi parameter algebraic collision rules were also proposed in prior work [35, 120]. These laws model phenomena like tangential compliance in addition to restitution and friction. They are employed primarily when the need for accuracy outweights the need for performance.

Algebraic collision laws are suitable for simulations with interactive performance because they do not require integration during collisions. However, their direct application to multiple collisions may result in energetically inconsistent results (i.e., an increase in the kinetic energy of the colliding bodies or dissipation of energy for perfectly elastic collisions), as pointed out in [26, 27, 78, 145]. Some event-driven constraint-based simulations avoid such difficulties by sequencing collisions according to various heuristics [35, 134]. However, collision sequencing is not suitable for haptics because it is not guaranteed to terminate in a finite number of steps [35]. Techniques for resolving multiple collisions simultaneously are needed. Such techniques were developed by formulating the dynamic response based on the complementarity rule and combining it with the time integration in time-stepping simulations. Time-stepping algorithms were developed based both on Newton's [112] and on Poisson's [6–8,37,104,142,143] restitution hypotheses. However, most existing time-stepping algorithms resolve the simultaneous collisions by iteration [7, 8, 104, 112, 142, 143], thus having no guaranteed completion time. Time-stepping methods with guaranteed completion time [6,37] have been proven not to increase the kinetic energy of the system only for perfectly plastic collisions between smooth convex objects [6]. The extension of these algorithms to arbitrary coefficients of restitution and polyhedral objects is an open research question.

This brief survey of restitution laws employed for impulse simulation indicates that only algebraic rules are suitable for algorithms with real time performance. Moreover, these rules must carefully be extended to simultaneous collisions to ensure that the kinetic energy of the system does not increase during collisions.

2.3 Haptic interaction within rigid multibody virtual environments

To enable haptic interaction with rigid multibody virtual worlds, prior haptics research either developed special-purpose dynamic response algorithms or employed methods already proposed in computational dynamics, robotics, and graphics. Specific force computation techniques were developed: (i) for point interaction with static collections of polyhedra [129,150,155]; (ii) for point interaction with static objects described by implicit functions [97,133]; (iii) for line interaction with static polyhedral environments [17]; and (iv) for rigid body interaction with static environments represented as voxel maps [103,123]. Realistic rigid body interactions within dynamic multibody environments were computed using the general-purpose simulation methods described in Section 2.1.2.

Most special-purpose techniques developed in haptics are penalty-based methods that achieve improved performance by taking advantage of the specific representation of the virtual environment [97,103,123,133] or the virtual tool [17] to speed up collision detection. The “god-object” [150,155] and the “virtual proxy” [129] techniques are constraint-based algorithms designed to overcome difficulties inherent in fixed step penalty-based simulations. They represent the user’s finger through a point (sphere) that follows the user’s motion as close as allowed by the objects in the virtual environment. These methods eliminate disadvantages inherent in fixed step penalty-based techniques. Namely, they prevent users from moving through thin objects in one simulation step and uniquely define a penetration distance at environment corners. In turn, users perceive constant environment stiffness throughout the interaction (while they perceive higher stiffness at concave corners and lower stiffness at convex ones in penalty-based simulations). However, the extension of the god-object and the virtual proxy methods to rigid body interaction is not straightforward [126]. Moreover, the perceived rigidity of virtual worlds simulated using these methods is similar to that of penalty-based environments. This is because god-object and virtual proxy simulations do not compute constraint-based forces and impulses and users feel only penalty-like forces applied to them by the interaction controller.

Haptic rendering of rigid body interaction with dynamic multibody virtual environments was enabled in prior work by employing all general-purpose simulation methods described in Section 2.1.2. The impulse-based technique did not adequately render contact and friction in an initial implementation [34] and was not pursued further. Penalty-based simulations are currently used in conjunction with advanced collision detection algorithms. These algorithms achieve real time performance through combining local and global search techniques [59, 75, 76, 83, 85] and eliminate artifacts due to discrete object representation through mesh decimation and mesh filtering methods [117]. Event-driven [22, 128, 149] and time-stepping [139] constraint-based simulations were used in haptics by bridging the gap between the virtual environment performance and the required haptic rate either through control or through a local model of interaction.

2.3.1 Haptic controllers

Control approaches aim to eliminate the instability caused by the (nonuniform) computational delay of the virtual environment through added virtual damping. They curb the energy flow from the virtual environment by using a "virtual coupler" [3, 42] (Figure 2.8). The virtual coupler is a stiffness and damping connection between the haptic interface and the virtual environment that limits the maximum impedance displayed by the device. It guarantees stable interaction with passive, nondelayed virtual environments [3]. Although the passivity and the nondelayed assumptions are quite restrictive, the virtual coupler is often used for enabling interaction with simulations with variable delay [22, 128, 138, 139]. Nonlinear damping was proposed in [65] to passivate the discrete implementation of virtual contacts and design guidelines were derived in [105] for adding virtual damping that guarantees stable interaction with both delayed and nondelayed nonlinear mass-spring-damper (i.e., penalty-based) environments.

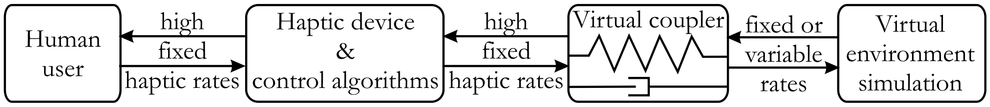


Figure 2.8: The virtual coupler.

One problem with adding virtual damping is that it degrades the transparency of the interaction by filtering out phenomena that rely on fast force transitions, such as collisions and stick-slip friction. In addition, the virtual coupler is a conservative design. The damping must be chosen such that stable interaction is guaranteed for the worst case interaction. A less conservative design was proposed in [95]. Specifically, for virtual environments with dynamics represented through several nonlinear functions, a virtual coupler was used for each function and stable switching between the various virtual couplers was devised based on passivity of hybrid systems arguments. However, stability is achieved by delaying the switches compared to the user commands, which, again, degrades transparency.

2.3.2 Local models of interaction

The local model of interaction approach decouples the force computation loop from the virtual world simulation. The decoupling alleviates the haptic rate demand on the simulation. Therefore, virtual environments of increased complexity can be rendered and multiple networked users may be allowed to manipulate the same virtual environment. However, realistic haptic feedback can be applied to users only if the setpoints of the control loop are suitable approximations of the forces acting on the virtual tool (i.e., the virtual object manipulated by the user). The techniques proposed for approximating the virtual interactions within rigid multibody virtual environments are presented in this section.

The decoupling of the force computation from the virtual environment was first proposed in [79], in a haptic extension of the X-Windows graphical user interface (GUI). A haptic model of the GUI, running on a microcontroller, was used to compute forces at a fixed control rate of 1kHz. The model included constraints corresponding to window borders, pull-down menus, cursor forces towards icons, etc., and was updated asynchronously by the X-Windows host according to the status of the GUI. A local model was not needed due to the relative simplicity of the virtual environment (few constraints, typically horizontal and vertical).

The earliest local model of force computation used in the control loop was proposed in [2] for point interaction within virtual environments. It included the position of the active constraint and its outward normal. Hence, it was a geometrical local model of the virtual environment. The local geometry shifted the computational delay of the simulation to delay in updating the local geometry. Therefore, larger contact stiffnesses were achievable, but force discontinuities arose at model updates due to geometry discontinuities. The force discontinuities introduced perceptual artifacts or destabilized the interaction.

Improvements to this geometric local model aim to preserve stability and eliminate the perceptual artifacts by applying continuous forces to users. In [149], kinematic constraints were proposed for diminishing force discontinuities due to point contact with dynamic objects. The kinematic constraints are geometric constraints that move locally with constant speed, equal to their speed in the virtual environment at the moment of the update. Hence, the local model uses a first order predictor to compute the constraint position between simulation updates. Rather than augmenting the local model, a force smoothing scheme based on linear interpolation between old and new constraints was designed in [100]. The most advanced local model of point interaction was presented in [119]. This model includes all virtual environment geometry that might be reached by the user until the next simulation update. Predicted geometry is selected based on the velocity of the user. This model was shown to ensure the continuity of the local geometry for user speeds of 0.2m/s in an environment with an average edge length of 7.5mm, typically running at 20Hz. One disadvantage of this model is that it is computationally expensive, at times barely finishing in one simulation step [119].

A local model of rigid body interaction, called the intermediate representation, was proposed in [21]. The intermediate representation includes all active constraints, enabling the user to feel crisp, stiff contacts. However, instability arose during tightly constrained motions of the virtual tool, such as during peg-in-hole insertion. Therefore, the intermediate representation was used to constrain only the translation of the haptic interface and a virtual coupler was employed to constrain its rotation.

Local models were also proposed for haptic interaction with deformable objects [9, 14, 96]. They aim to accurately render variations in model stiffness to users. Hence, these models are not suitable for rigid body interaction, which requires users to feel constant environment stiffness, but varying inertia and restrictions imposed on their motion by other virtual objects.

Increased transparency can be achieved within local models of interaction if continuity is ensured at updates. This is because users can feel stiffer contacts, impacts, and dry friction. A synopsis of the techniques employed for rendering impacts and friction to users is presented in the following sections.

2.3.3 Haptic rendering of impact

A key factor in enhancing the realism of the haptic manipulation of rigid multibody virtual worlds is the perceived contact rigidity. As demonstrated by psychophysical studies [90,125], the perceived rigidity of virtual contacts can be improved through applying large forces to users when new contacts arise. However, little haptics work exists that enhances the perception of rigidity through applying large forces to users when new contacts arise.

The earliest large forces rendered to users are the “braking pulses” [131]. The braking pulses arise from a virtual wall model with high initial contact damping. They are designed to dissipate the entire kinetic energy of the haptic device during point interaction within virtual environments. Hence, the braking pulses can also be interpreted as impulsive forces arising due to a perfectly plastic collision. They are shown to improve both the stability and the perceived rigidity of the contact.

Large forces applied on users upon contact during rigid body manipulation of virtual worlds were physically motivated either as "pre-contact braking forces" [103] or as impulsive forces [43,44]. The pre-contact braking forces arise from the interaction with a viscous layer surrounding the virtual objects and dissipate the translational kinetic energy of the virtual tool along the direction of approaching contact. The impulsive forces arise from a planar model of rough impact employing on Poisson's restitution rule. Simultaneous collisions satisfy this model in a least squares sense and their passivity is not proven.

Recently, open loop force pulses [71] were used to increase the perceived rigidity of surfaces. In addition to canceling user's momentum upon point contact with a rigid environment, the open loop pulses generate high frequency vibrotactile content that approximates the frequency content of contact with real surfaces.

2.3.4 Haptic rendering of friction

Since friction is ever-present in the physical world, realistic haptic manipulation of virtual environments requires it to be suitably simulated and applied to users. Several friction rendering techniques exist in haptics, all of them approximating Coulomb (dry) friction. Of course, Coulomb friction is only a macroscopic approximation of microscopic phenomena. Nonetheless, user studies [124] have shown that haptic rendering of dry friction affects task performance in a manner similar to that of real friction.

Haptic rendering of friction is based on the simulation models proposed by Dalh [45], Karnopp [77], and Haessig and Friedland [61]. Various modifications of the Karnopp model were employed in [113,124,131,155]. For example, viscous friction replaced kinetic friction in [131], the sticking region was designed based on the characteristics of the human finger pad in [113], and both kinetic and viscous friction were used in [124]. The bristle model proposed by Haessig and Friedland was used to simulate both friction and adhesion in [36]. Lastly, a modification of the Dahl model was proposed in [67] that eliminates the spurious position drift of the original model. Additionally, the modified model depends only on position, which makes it robust to noise and suitable for implementation in event-based simulations with non-uniform sampling. An implementation of the modified Dahl model for haptic interaction with a virtual environment through a virtual tool was reported in [85].

2.3.5 Summary

Rigid multibody modeling and simulation techniques with interactive performance have been developed in computational mechanics, robotics, and graphics research. The potential of these techniques to meet the specific demands of haptic manipulation of virtual environments (guaranteed real time performance for stable and transparent interaction, and physical realism for meaningful force feedback) has been considered. The investigation has shown that constraint-based and impulse-based techniques represent the physical interaction more accurately than penalty-based techniques, because they enforce exact contact rigidity. However, perfectly rigid contacts are compatible with the fixed step haptic integrators only if multiple collisions are resolved simultaneously and a constraint stabilization mechanism is available. In existing research, only constraint-based simulations of virtual environments with perfectly plastic contacts have been developed that are compatible with fixed step integrators.

Haptic multibody simulation techniques, approaches to guaranteeing the force computation rates in arbitrarily complex virtual environments, and methods for increasing the realism of the interaction through impact and friction rendering have also been overviewed. Regardless of how the virtual world is generated, haptic force update rates are achieved in prior research by connecting the simulation to the device through a virtual coupler, i.e., by applying penalty-like forces to users through control. Decoupling of the simulation from the control loop through a local model of interaction has been demonstrated only for point interaction, while impact-like forces have been applied only to users interacting with planar virtual environments.

A novel haptic simulation approach that allows users to feel impacts during their rigid body interaction within spatial virtual environments is presented in the following chapter. Its implementation in a local model of rigid body interaction is detailed in Chapter 6.

Chapter 3

Experimental setup

This chapter presents the haptic interaction system that will be used throughout this thesis to validate newly proposed simulation and control techniques that enable efficient and transparent haptic rendering of rigid body motion with constraints.

All experiments described in later chapters use the planar haptic simulation system depicted in Figure 3.1. This sytem comprises: a haptic device together with its input/output (I/O) interfaces and power amplifiers [136]; a haptics (real time) processor; and a graphics (host) processor with a graphical display. The haptic device is a 3 DOF interface that has a workspace larger than 100mm × 100mm and allows unlimited rotation. Hence, it enables planar rigid body haptic manipulations. The haptics processor is a 700MHz Pentium III personal computer running VxWorks^{TM}, while the graphics processor is a 2.4GHz Pentium IV personal computer running Windows2000^{TM}. The two processors communicate via a UDP socket. As depicted in Figure 3.2, the communication between the virtual environment simulation and the force control loop is implemented using two software architectures.

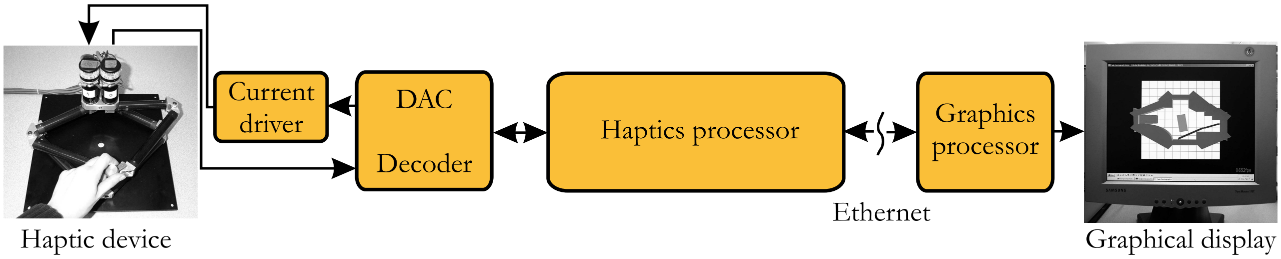


Figure 3.1: The planar haptic simulation system used for experiments throughout this thesis.

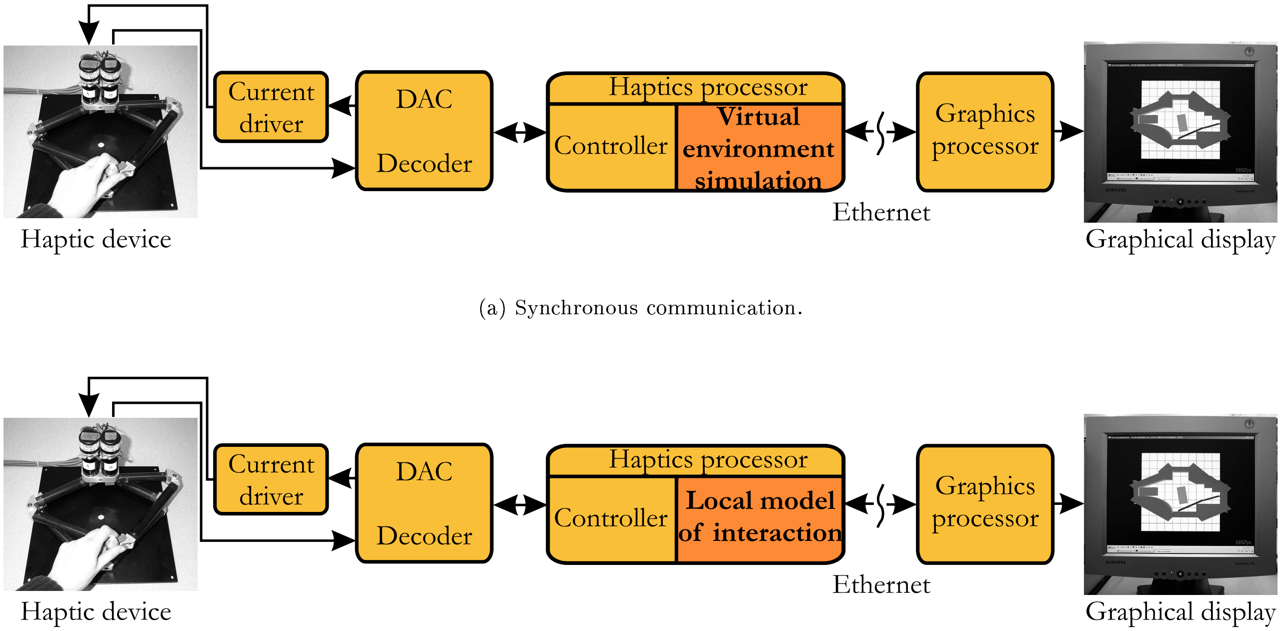
Synchronous communication is implemented by running both the virtual environment and the device control on the haptics processor (see Figure 3.2(a)), at a frequency of 512Hz. In this architecture, the graphics processor only displays the virtual environment, at a frequency of approximately 30Hz. The graphics processor acts as the server and polls the socket for new data asynchronously, each time it has finished displaying the old data. The haptics processor serves as the client and updates the socket at the end of each control step.

Asynchronous communication is implemented by running the force control loop and the local model of interaction proposed in Chapter 6 on the haptics processor at 512Hz (see Figure 3.2(b)). In this architecture, the graphics processor both generates and displays the virtual environment, at frequencies varying between 30Hz and 60Hz. The haptics processor acts as the server. Hence, the local model polls the socket for new data at the beginning of each control step. When new data are available, the local model updates its state and acknowledges the receipt of the packet by sending back the proxy state. The virtual environment sends packets asynchronously, each time it has completed a simulation step. Vortex^{TM}, a physics based engine developed by CMLabs Simulations Inc.[1], is used to generate the virtual environment in the asynchronous communication architecture.

The force control loop comprises two components: the haptic interaction controller and the device controller. The haptic interaction controller [136] implements a four channel teleoperation architecture [89]. In the synchronous communication paradigm, the four channel controller transmits wrenches (i.e., forces and torques) between the device and the virtual object manipulated by the user (hereafter called virtual tool), and coordinates positions between them. In the asynchronous communication paradigm, the four channel controller transmits wrenches between the device and a proxy of the virtual tool in the local model of interaction, and coordinates positions between the haptic interface and the proxy. The impedance device controller [136] shapes the dynamics of the haptic interface to match those of the virtual tool or of the proxy when the synchronous or the asynchronous communication architectures are used, respectively. Both controllers are presented in further detail in Appendix D.

All experiments later described in this thesis use the planar haptic simulation system described in this chapter and the synchronous and the asynchronous testbed virtual environments depicted in Figures 3.3(a) and 3.3(b), respectively. In both virtual environments, the moving objects have mass $m = 2$kg and moment of

[1] www.criticalmasslabs.com



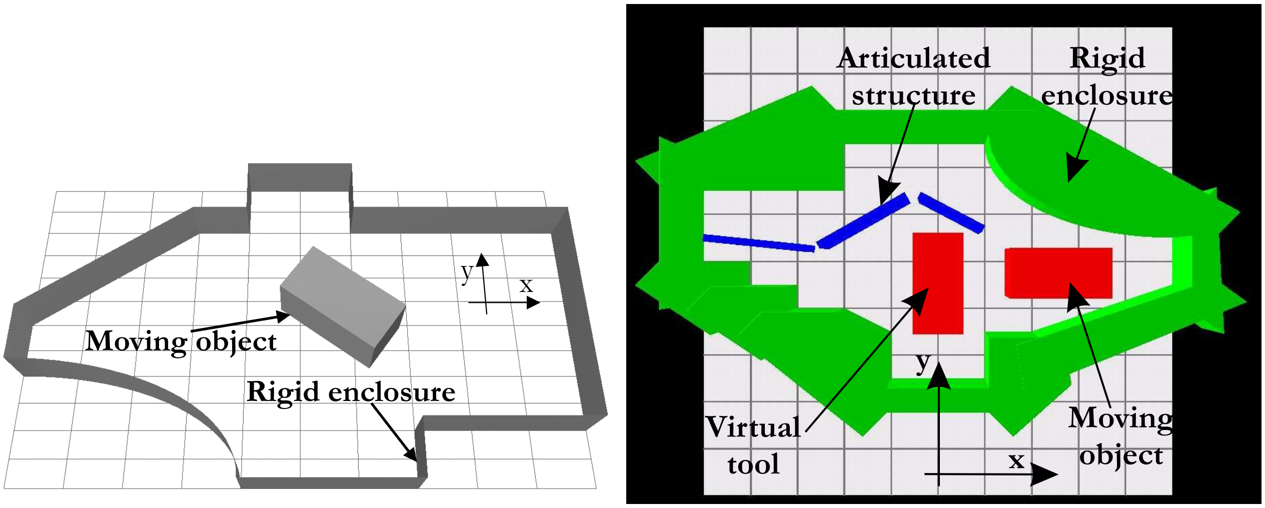
(a) Synchronous communication.

(b) Asynchronous communication.

Figure 3.2: Communication between the virtual environment simulation and the force control loop in the planar haptic interaction system used in this thesis.

inertia $I = 0.005\text{kg}\cdot\text{m}^2$. The links of the virtual linkage in Figure 3.3(b) are numbered from the proximal link to the distal one. They have masses $m_1 = 3\text{kg}$, $m_2 = 3\text{kg}$, $m_3 = 1\text{kg}$, moments of inertia $I_1 = 0.015\text{kg}\cdot\text{m}^2$, $I_2 = 0.015\text{kg}\cdot\text{m}^2$, $I_3 = 0.005\text{kg}\cdot\text{m}^2$, and lengths $l_1 = 4.2\text{cm}$, $l_2 = 4.2\text{cm}$, $l_3 = 3\text{cm}$, respectively.

Several experiments presented in later chapters require performance comparisons between the simulation and control methods developed in this thesis and prior approaches. Since users are not able to apply the exact same interaction wrenches during successive trials, these experiments cannot involve manipulations performed by humans. To enable comparison, these experiments are controlled experiments. The same initial conditions and the same "user" are ensured during one experiment through replacing the user's hand by controlled wrenches. Since the haptic device is an impedance-type interface, elimination of the adaptive damping associated with the user manipulation of the device represents a worst-case scenario for stability [64].



(a) Virtual environment synchronous with the haptic loop.

(b) Virtual environment asynchronous with the haptic loop.

Figure 3.3: Testbed virtual environments used in the experiments.

Chapter 4

Haptic rendering of rigid contact

An *impulse-augmented penalty simulation* approach is proposed in this chapter that enables users to feel collisions during their haptic manipulation of rigid multibody virtual environments. The chapter starts with a synopsis of the proposed approach and of the haptic controller that applies the simulated interactions to users. Then, the rigid body contact model underlying the impulse-augmented penalty approach is discussed. The dynamics employed during contact are subsequently presented. These dynamics include a friction model developed in computational dynamics [61] that is for the first time employed for haptic rendering of dry friction, and is compared through simulations and experiments to existing models. The dynamics used upon contact follow. Next, the performance of the impulse-augmented penalty simulation approach is compared to the performance of existing approaches through simulations. Limitations imposed by the haptic device are discussed and a solution is devised to address them. Lastly, experiments are presented to validate the simulation results presented in earlier section. Coordinate invariant representation of contact at the user's hand is deferred to the following chapter.

The performance of the impulse-augmented penalty simulation approach is contrasted to the performance of prior approaches through investigating the realism and the stability of the haptic interaction. The realism of the interaction is assessed through evaluating the transient response of the closed loop system that comprises the user, the device, and the virtual environment simulation. A closed loop system with smaller overshoot and shorter settling time represents more realistic interaction. This is because smaller overshoot means that users violate the virtual constraints less, and shorter settling time means that users stop faster upon contact with a static virtual environment. In other words, the trajectory imposed on users by the virtual constraints is closer to the trajectory that a physical rigid environment would impose on them. The stability of the haptic interaction is defined as the coupled stability (in the sense of limited velocities and forces [39]) of the closed loop system that comprises the user, the device, and the virtual environment simulation. Since a feedback interconnection of passive systems is necessarily stable [148] and the haptic device and the user are passive [69], the passivity of the simulation guarantees the coupled stability of the haptic interaction. Furthermore, enhanced passivity of the virtual environment enhances the passivity of the closed loop system, thereby improving the stability of the haptic interaction. Therefore, interaction stability is assessed in this thesis through examining the passivity of the virtual contacts by monitoring the kinetic energy of the user's hand during the interaction.

The schematic of haptic interaction within rigid multibody virtual environments generated using the methods introduced in this chapter is shown in Figure 4.1. In this figure, $\mathbf{F}_h$ is the user-applied wrench (force and torque), $\mathbf{F}_{env}$ is the environment wrench represented at the user's hand, and $\mathbf{f}_1$ and $\mathbf{f}_2$ are contact forces between the virtual tool and other virtual objects. Furthermore, $\mathbf{x}_h$ and $\dot{\mathbf{x}}_h$ are the body position (position and orientation) and the body velocity (linear and angular velocity) of the user's hand, and $\mathbf{x}_{sim,h}$ and $\dot{\mathbf{x}}_{sim,h}$ are simulated body position and body velocity of the user's hand.

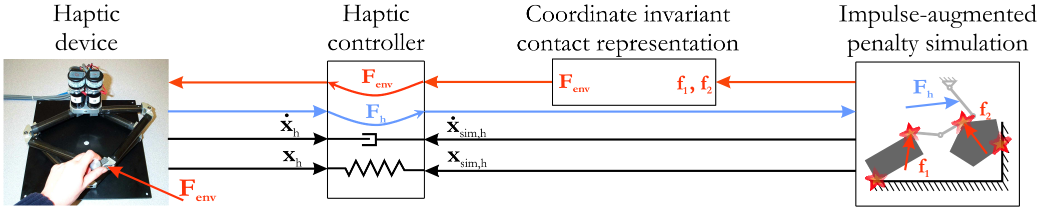


Figure 4.1: Haptic manipulation in an impulse-augmented penalty virtual world.

4.1 Synopsis of the impulse-augmented penalty simulation and of the haptic controller

The impulse-augmented penalty simulation approach is motivated by user studies [90, 125] that demonstrate that the perceived rigidity of virtual contacts can be improved by applying large forces to users upon contact. The approach computes impulsive forces upon contact and then penalty and friction forces during contact. The impulsive forces are derived using a new collision resolution method that never increases the kinetic energy of the simulated system. When new contacts arise, the impulsive forces generate large hand accelerations without requiring increased contact stiffness and damping.

The approach generalizes earlier work in [131] in two ways: (i) it allows the energy dissipated upon contact to be adjusted through the coefficient of restitution; and (ii) it is suitable for rigid body manipulation, as opposed to point interaction. Compared to earlier work in [43] and [44], it uses a new contact model and Newton's restitution hypothesis for collision resolution. The contact model allows transitions to collision from all contact states. Therefore, rigid body contact is represented more accurately (see Section 4.2) and the simulation can account for device limitations (see Section 4.5.2). Newton's restitution law allows multiple collisions to be resolved such that the kinetic energy of the system does not increase during collisions. Manipulation of both virtual objects and linkages is enabled through simulating the dynamics of the virtual environment in configuration space. In this space, virtual objects and linkages are represented as points and no distinction between them is necessary.

Ultimately, transparent haptic interaction is achieved by applying the simulated contact forces to users through a suitable haptic controller. The controller used for interaction with impulse-augmented penalty virtual worlds is a four channel teleoperation controller [136]. It coordinates both positions and forces between the virtual tool and the haptic device, as shown in Figure 4.1. The two position coordination channels implement a generalized (translational and rotational) spring-damper connection (i.e., a proportional derivative - PD - controller) between the virtual tool and the device, maintaining kinematic correspondence and eliminating drift. The two force coordination channels apply the hand wrench (force and torque) $\mathbf{F}_h$ to the virtual tool in the simulation, and the environment wrench to the user. Unlike haptic controllers that employ only position coordination and apply just penalty-like wrenches to the user's hand [3, 42], the four channel teleoperation controller directly applies the simulated impulsive interactions to users through its force channels. Hence, it enables users to feel collisions and improves the stability and the perceived rigidity of the virtual contacts.

Figure 4.2 illustrates the behavior of the impulse-augmented penalty simulation: upon contact (Figures 4.2(b) and 4.2(d)), the rigid body contact model exactly enforces contact rigidity, while during contact (Figures 4.2(c) and 4.2(e)), contact rigidity is only approximated.

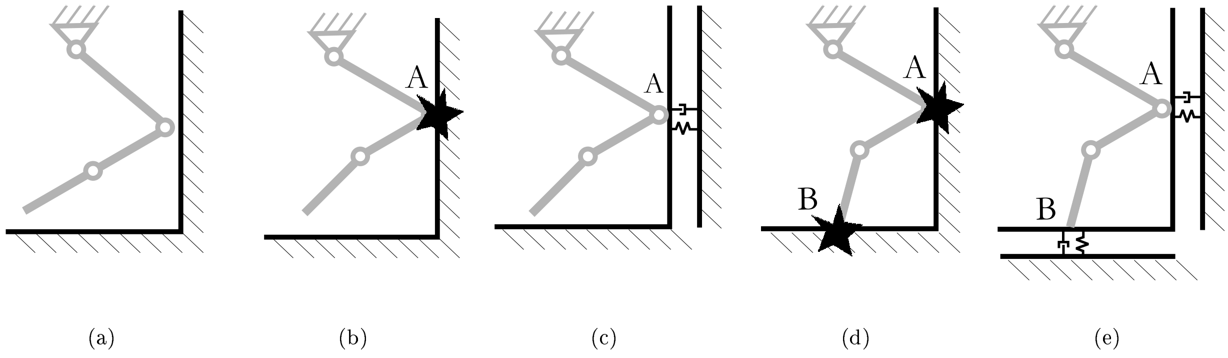
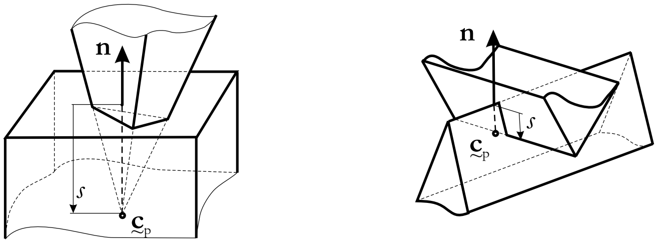


Figure 4.2: The impulse-augmented penalty-based rigid body contact model: contact rigidity is exactly enforced upon contact and only approximately enforced during contact.

4.2 The contact model

The impulse-augmented, penalty-based, rigid-body-contact model is a dynamic model rather than a geometric one. It is used for computing interaction forces and impulses, based on the geometric information provided by a collision detection algorithm that is implemented in the virtual environment. Typically, the collision detection algorithm decomposes each rigid object into a collection of convex polyhedra and computes contacts between pairs of these polyhedra [85]. In three dimensional (3D) virtual environments, these contacts represent either vertex-face or edge-edge contacts[1], Figure 4.3. For each contact, it provides a contact point $\underset{\sim}{\mathbf{c}}_p$, a penetration depth (equal and opposite to the separation distance s between the bodies), and a contact normal direction $\mathbf{n}$.


n
s
$\underset{\sim}{\mathbf{c}}_p$
n
s
$\underset{\sim}{\mathbf{c}}_p$

(a) Vertex-face contact. (b) Edge-edge contact.

Figure 4.3: Contact information received from collision detection.

[1] All other types of contacts, for example face-face contacts, are represented as a finite number of vertex-face and/or edge-edge contacts [152].

In the impulse-augmented penalty simulation approach, a *contact* is defined by this geometric information plus the contact velocity $\mathbf{v}$. The contact velocity is the relative velocity between the contacting polyhedra at the contact point. It is defined such that the normal contact velocity v_n (i.e., the component of the contact velocity along the contact normal $v_n = \mathbf{n}^T \cdot \mathbf{v}$) is negative if the polyhedra move into each other. A *penetrating contact* is a contact with negative normal contact velocity $v_n < 0$. A *separating contact* is a contact with nonnegative normal contact velocity $v_n \geq 0$.

Given a rigid multibody virtual environment, two rigid bodies are said to be in the same *contact group* if there exists a chain of contacting moving rigid bodies[2] between them [122]. The *rigid body contact model* has three states: free motion, colliding contact, and resting contact. A rigid body is said to be in *free motion* if it has no contacts. A rigid body is said to be in *colliding contact* if at least one new penetrating contact exists within its contact group. Finally, a rigid body is said to be in *resting contact* if it is neither in free motion nor in colliding contact. Note that a body can have non-zero acceleration and velocity during “resting contact”. This terminology is used in the present work to maintain consistency with prior literature on the subject [152].

Note that, due to the fixed time step of the haptic simulation, bodies may transition from free motion to resting contact. An example is depicted in Figure 4.4, where contact A does not exist at time t and is already a separating contact at time $t+1$. Hence, collisions might be missed in an impulse-augmented penalty virtual environment. Note also that bodies may remain in colliding contact for multiple consecutive simulation steps because new penetrating contacts may appear at every step.

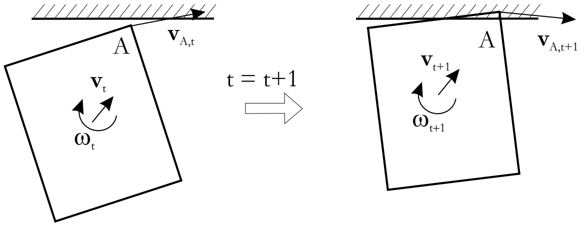


Figure 4.4: Example transition from free motion to resting contact during a fixed step haptic simulation. $\boldsymbol{\omega}$ is the angular velocity of the body, $\mathbf{v}$ is the velocity of its center of mass, $\mathbf{v}_A$ is the velocity of point A on the body, and t is the time step of the simulation.

By including the colliding contact state, the impulse-augmented penalty-based rigid body contact model approximates rigidity better than the penalty-based model. The tradeoff is that, by allowing body interpene-

[2] as opposed to static bodies, i.e., rigid walls.

tration during resting contact, the model provides a poorer approximation of rigidity than the constraint-based contact model. The combination of approximate and exact constraint enforcement requires penalty-based resting contact dynamics and constraint-based collision resolution. The numerical methods employed to simulate resting contact are presented in the following section. The simultaneous resolution of multiple collisions is discussed in Section 4.4.

4.3 Resting contact

This work considers the case in which users manipulate both virtual objects and virtual linkages (i.e., the virtual tool can be a single object or a chain). Realistic forces during these types of interaction can be computed by representing the virtual world dynamics either in Cartesian space or in configuration space. In Cartesian space, constraint equations must be added to maintain the bilateral constraints and the simulation must integrate a computationally expensive differential algebraic system of equations for which constraint satisfaction may be problematic. In configuration space, the bilateral constraints are embedded in the coordinate representation. In this case, only a reduced number of coordinates must be integrated and bilateral constraint satisfaction is guaranteed. Therefore, configuration space dynamics are used in the proposed simulation to compute the interactions between the virtual tool and the virtual world.

Since bilateral constraints are incorporated in the coordinate representation, only contact and user applied forces must be included in the dynamics equations. Consider a contact group with d degrees of freedom (DOFs) and c resting contacts. In configuration space, its dynamics are:

$$\mathbf{D}\left(\mathbf{q}\right)\ddot{\mathbf{q}} + \mathbf{B}\left(\mathbf{q},\dot{\mathbf{q}}\right) + \mathbf{G}\left(\mathbf{q}\right) = \sum_{i=1}^{c} \mathbf{J}_i^T\left(\mathbf{q}\right)\mathbf{f}_i + \mathbf{J}_h^T\left(\mathbf{q}\right)\mathbf{F}_h. \tag{4.1}$$

In Equation (4.1), $\mathbf{D}\left(\mathbf{q}\right) \in \mathcal{R}^{d\times d}$ is the configuration space inertia matrix of the contact group, $\mathbf{B}\left(\mathbf{q},\dot{\mathbf{q}}\right) \in \mathcal{R}^d$ represent Coriolis and centripetal effects, $\mathbf{G}\left(\mathbf{q}\right) \in \mathcal{R}^d$ are the gravitational terms, $\mathbf{J}_i\left(\mathbf{q}\right) \in \mathcal{R}^{3\times d}$ is the Jacobian computed at the i-th contact, $\mathbf{f}_i \in \mathcal{R}^3$ is the Cartesian space contact force at the i-th contact, $\mathbf{J}_h\left(\mathbf{q}\right) \in \mathcal{R}^{6\times d}$ is the Jacobian computed at the user's hand, $\mathbf{F}_h = \begin{pmatrix}\mathbf{f}_h^T & \boldsymbol{\tau}_h^T\end{pmatrix}^T \in \mathcal{R}^6$ is the wrench applied by the user (with $\mathbf{f}_h \in \mathcal{R}^3$ being the user-applied force and $\boldsymbol{\tau}_h \in \mathcal{R}^3$ the user-applied torque), and $\mathbf{q} \in \mathcal{R}^d$, $\dot{\mathbf{q}} \in \mathcal{R}^d$, and $\ddot{\mathbf{q}} \in \mathcal{R}^d$ are the configuration space positions, velocities, and accelerations, respectively (see Figure 4.5).

If the contact group consists of both virtual objects and virtual linkages, the matrices and vectors in

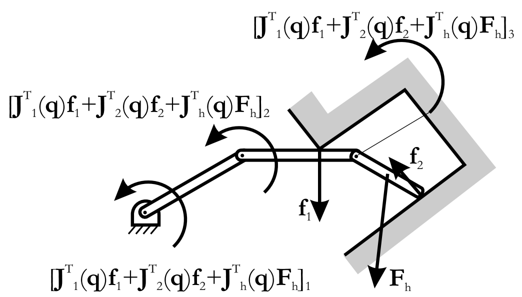

$[\mathbf{J}^T_1(\mathbf{q})\mathbf{f}_1+\mathbf{J}^T_2(\mathbf{q})\mathbf{f}_2+\mathbf{J}^T_h(\mathbf{q})\mathbf{F}_h]_3$
$[\mathbf{J}^T_1(\mathbf{q})\mathbf{f}_1+\mathbf{J}^T_2(\mathbf{q})\mathbf{f}_2+\mathbf{J}^T_h(\mathbf{q})\mathbf{F}_h]_2$
$\mathbf{f}_2$
$\mathbf{f}_1$
$[\mathbf{J}^T_1(\mathbf{q})\mathbf{f}_1+\mathbf{J}^T_2(\mathbf{q})\mathbf{f}_2+\mathbf{J}^T_h(\mathbf{q})\mathbf{F}_h]_1$
$\mathbf{F}_h$

Figure 4.5: Example contact forces and and hand wrenches arising during the haptic manipulation of a contact group with dynamics computed in configuration space.

Equation (4.1) are obtained by concatenating the matrices and vectors corresponding to each object and linkage. For example:

$$\mathbf{q}^T = \begin{pmatrix} \mathbf{q}_1^T & \cdots & \mathbf{q}_m^T \end{pmatrix} \tag{4.2}$$

and:

$$\mathbf{D}\left(\mathbf{q}\right) = \begin{bmatrix} \mathbf{D}_1\left(\mathbf{q}_1\right) & \cdots & \mathbf{0} \\ \vdots & \ddots & \\ \mathbf{0} & \cdots & \mathbf{D}_m\left(\mathbf{q}_m\right) \end{bmatrix}. \tag{4.3}$$

In Equation (4.3), m is the total number of virtual objects and virtual linkages and the configuration space dynamics of a virtual rigid body are the same as its Cartesian space dynamics.

The contact forces in Equation (4.1) have a component $f_{n,i}$ along the contact normal direction $\mathbf{n}_i$, modeling contact rigidity, and a component $f_{f,i}$ along the direction $\mathbf{t}_i$ (orthogonal to the contact normal), modeling dry friction:

$$\mathbf{f}_i = f_{n,i}\mathbf{n}_i + f_{f,i}\mathbf{t}_i. \tag{4.4}$$

Resting contact is enforced using penalties. Hence, the normal component of the contact force at the i-th contact is computed by:

$$f_{n,i} = -K_{contact}s_i\left(\mathbf{q}\right) - B_{contact}v_{n,i}\left(\mathbf{q}\right). \tag{4.5}$$

In Equation (4.5), $K_{contact}$ and $B_{contact}$ are the contact stiffness and damping, s_i is the separation between bodies at contact (because bodies overlap, s_i is negative and equal to the penetration depth of the contact), and $v_{n,i}$ is the normal contact velocity. Dry friction can be simulated using any friction model that employs only local contact state information, such as [67] or [113]. However, the reset-integrator model proposed by Haessig and Friedland [61] is implemented in this work, for reasons discussed in the following section.

4.3.1 Friction modeling

The reset-integrator model was first used in haptics in [43]. According to this model, the friction force at each contact is computed by:

$$f_f = \begin{cases} (1+a)K_r z + \beta v_t & \text{if stick} \\ K_r z & \text{if slip} \end{cases}, \tag{4.6}$$

and the input to the integrator is:

$$\begin{aligned} \dot{z} = 0 \quad \text{if} & \begin{cases} v_t > 0 \quad \text{and} \quad z \geq z_0 \\ \qquad\quad \text{or} \\ v_t < 0 \quad \text{and} \quad z \leq -z_0 \end{cases} & \text{(slip)} \\ \dot{z} = v_t \qquad & \text{otherwise.} & \text{(stick)} \end{aligned} \tag{4.7}$$

In Equation (4.6), z is the strain of the bond between the contacting bodies, z_0 is the breakaway distance (i.e., the maximum strain of the bond), K_r is a spring stiffness, a is the "stiction gradient", v_t is the sliding velocity (i.e., the projection of the contact velocity on the contact plane $v_t = \mathbf{t}^T\mathbf{v}$), and β is a damping coefficient. In a haptics implementation, the model may exhibit chattering because of noisy velocity measurements. This difficulty is avoided by implementing a dead band of width $2v_{min}$ around zero sliding velocity. Specifically, Equation (4.7) is replaced by:

$$\begin{aligned} \dot{z} = 0 \quad \text{if} & \begin{cases} v_t > -v_{min} & \text{and} \quad z \geq z_0 \\ & \text{or} \\ -v_{min} < v_t < v_{min} & \text{and} \quad z \leq -z_0 \end{cases} & \text{(slip)} \\ \dot{z} = v_t \qquad & \text{otherwise.} & \text{(stick)} \end{aligned} \tag{4.8}$$

In the reset-integrator model, the contacting points on the two bodies are connected by a spring with stiffness $(1+a)K_r$ during sticking and K_r during sliding. A stiction gradient $a = 0$ describes contacts which have a coefficient of static friction μ_s equal to the coefficient of kinetic friction μ_k. A contact transitions from stick to slip when the strain of the bond exceeds the maximum strain z_0 and tends to increase further. Otherwise, the contact transitions to slip. The oscillations occurring when entering stiction are damped by the βv_t term.

The role of the damping factor in controlling the transition from slippage to stiction is illustrated through simulating the peg-in-hole manipulation depicted in Figure 4.6. In the simulation, the peg has mass $m = 2$kg and barely fits the hole, the normal contact force at each contact is $f_n = 0.175$N, and all contacts have the same coefficients of static $\mu_s = 0.25$ and kinetic $\mu_k = 0.2$ friction. The breakaway distance is $z_0 = 0.01$mm and the velocity dead band is $v_{min} = 0.1$mm/s. To match the experiments, the simulation time step is chosen $\frac{1}{512}$s. The simulated position of the COM of the peg, the sliding velocity at each contact (equal to the velocity of the COM of the peg), and the resultant friction force on the peg are shown in Figure 4.7 for three values of the damping coefficient: (i) no damping, $\beta = 0$; (ii) critical damping, $\beta = 118$; and, (iii) overdamping, $\beta = 1180$. For clarity, the figures depict the sliding velocity and the friction force only during the first 1.5 seconds of the first stiction period.

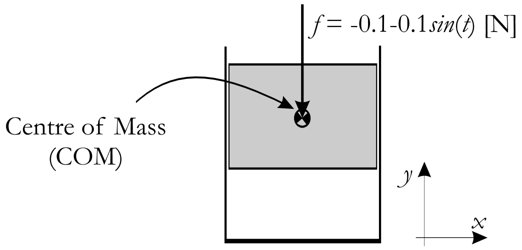
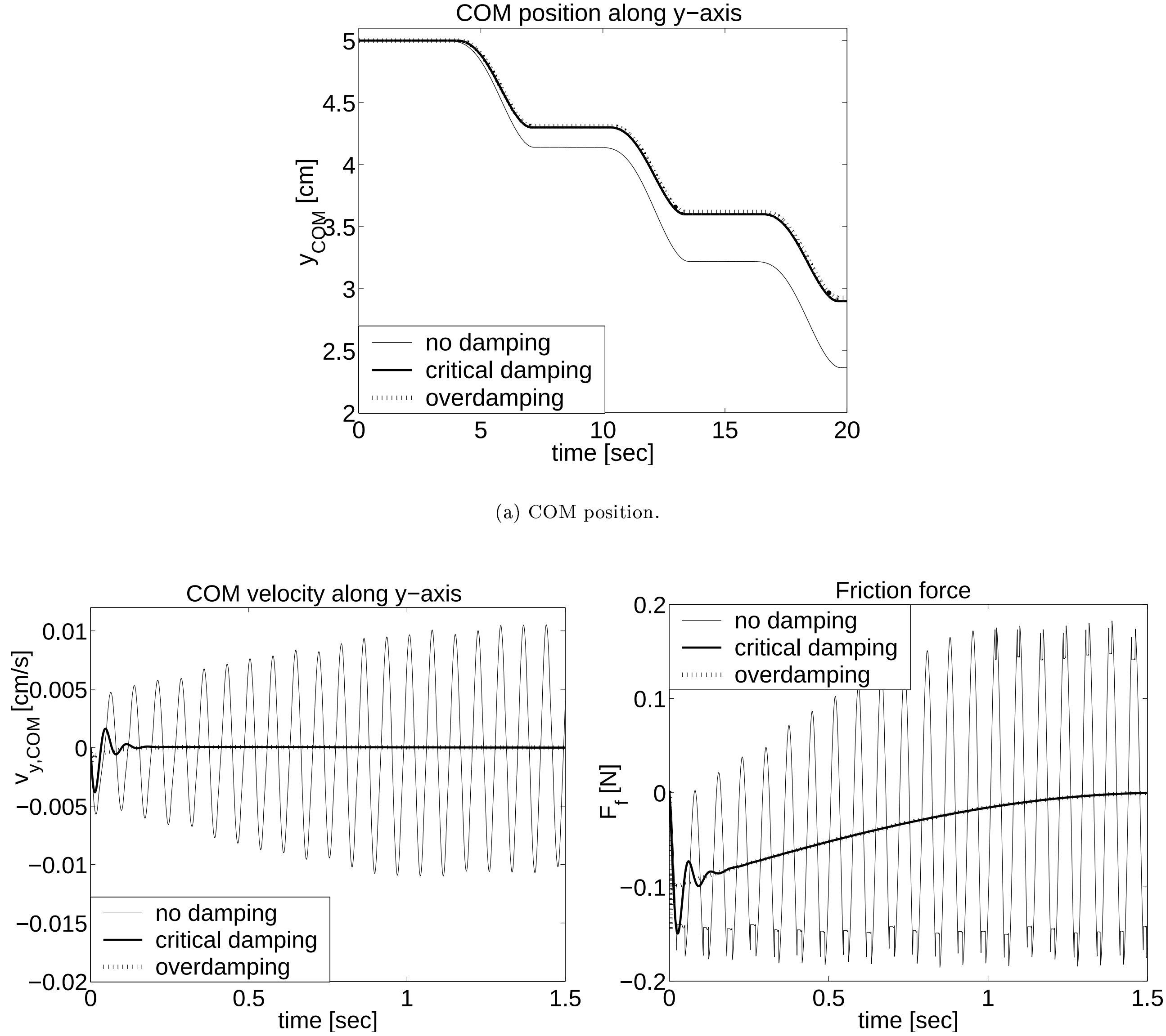


Figure 4.6: Simulated peg-in-hole manipulation.

The results in Figure 4.7 illustrate that the reset-integrator model successfully renders the slip-stick phenomenon regardless of damping. In addition, underdamping results in oscillatory sliding velocity and friction force during stiction. Critical damping and overdamping eliminate these oscillations when transitioning from slippage to stiction. In haptics, the friction force oscillations may be used for changing the feel of the contact. Psychophysical studies are required to clarify the relationship between the model damping and the user's perception of the contact characteristics.

By representing stiction through compliance in the contact plane, the reset-integrator model is similar to other haptic dry friction simulation techniques reviewed in Section 2.3.4, such as the model based on the human finger pad characteristics proposed by Nahvi *et al.* [113] and the modified Dahl method proposed by Hayward



(a) COM position.

(b) Sliding velocity.

(c) Resultant friction force.

Figure 4.7: Peg-in-hole simulation. Friction is simulated using the reset-integrator model [61] with various damping coefficients.

and Armstrong [67]. Like these methods and unlike the classical approximation of Coulomb friction (Figure 4.8), the model does not exhibit drift. In addition, it can either render oscillations at the transitions from slippage to stiction similar to the models proposed by Nahvi *et al.* [113] and by Hayward and Armstrong [67], or it can eliminate these oscillations similar to the classical Coulomb model. This is illustrated through simulating the manipulation shown in Figure 4.6 using the critically damped reset-integrator technique, the models proposed by Nahvi *et al.* and Hayward and Armstrong, and the classical approximation for implementing friction.

In the model based on the human finger pad characteristics [113], the slipping friction force $\mathbf{f}_f$ is computed by:

$$\mathbf{f}_f = \mu_d ||\mathbf{n}|| \frac{\mathbf{v}_t}{||\mathbf{v}_t||}, \tag{4.9}$$

where μ_d is the coefficient of dynamic frcition, $||\mathbf{n}||$ is the magnitude of the normal force, and $\mathbf{v}_t$ is the tangential velocity of the haptic interface. Stiction begind when $||\mathbf{v}_t||$ becomes smaller than a threshold velocity v_{min}. During stiction, the haptic interface is trapped in a circle of radius $\frac{\mu_d}{k_f}$ centered at $\mathbf{c}$ by a spring of stiffness k_f. When the interace transitions from slippage to stiction, the stiction centre $\mathbf{c}$ is computed by:

$$\mathbf{c} = \mathbf{a} - \frac{\mu_d}{k_f} \frac{\mathbf{v}_t}{||\mathbf{v}_t||}, \tag{4.10}$$

where $\mathbf{a}$ is the position of the haptic interface. The spring has a rupture limit of $\mu_s||\mathbf{n}||$ and slipping begins when the haptic interface is $\frac{\mu_d}{k_f}$ away from the stiction centre.

In the modified Dahl model [67], the dry friction force $\mathbf{f}_f$ is computed by:

$$\mathbf{f}_f = \mu_d ||\mathbf{n}|| \frac{\mathbf{v}_t}{||\mathbf{v}_t||} \frac{z}{z_{max}}, \tag{4.11}$$

where z is an internal position variable representing the distance from the sticking centre to the haptic interface and z_{max} is the maximum value of this distance. Stiction occurs when $z < z_{max}$, z is updated using forward Euler integration, and the time rate of change of z is computed by:

$$\dot{z} = \begin{cases} v_t & \text{if stiction, i.e., } ||z|| < z_{max} \\ v_t \left(1 - \frac{z}{z_{max}} sign(v_t)\right) & \text{if slippage, i.e., } ||z|| \geq z_{max} \end{cases}. \tag{4.12}$$

The simulation results are depicted in Figure 4.9. For clarity, the sliding velocity (equal to the peg COM velocity) and the friction force are plotted only during the first 1.5 seconds of the first stiction period.

The damping included in the reset-integrator model allows an extra degree of freedom in designing the contact characteristics: the higher the damping, the faster the oscillations occuring at the transition between slip and stick are suppressed. Therefore, this model is considered most suited for force feedback and proposed for use in haptics applications.

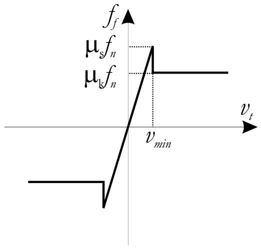
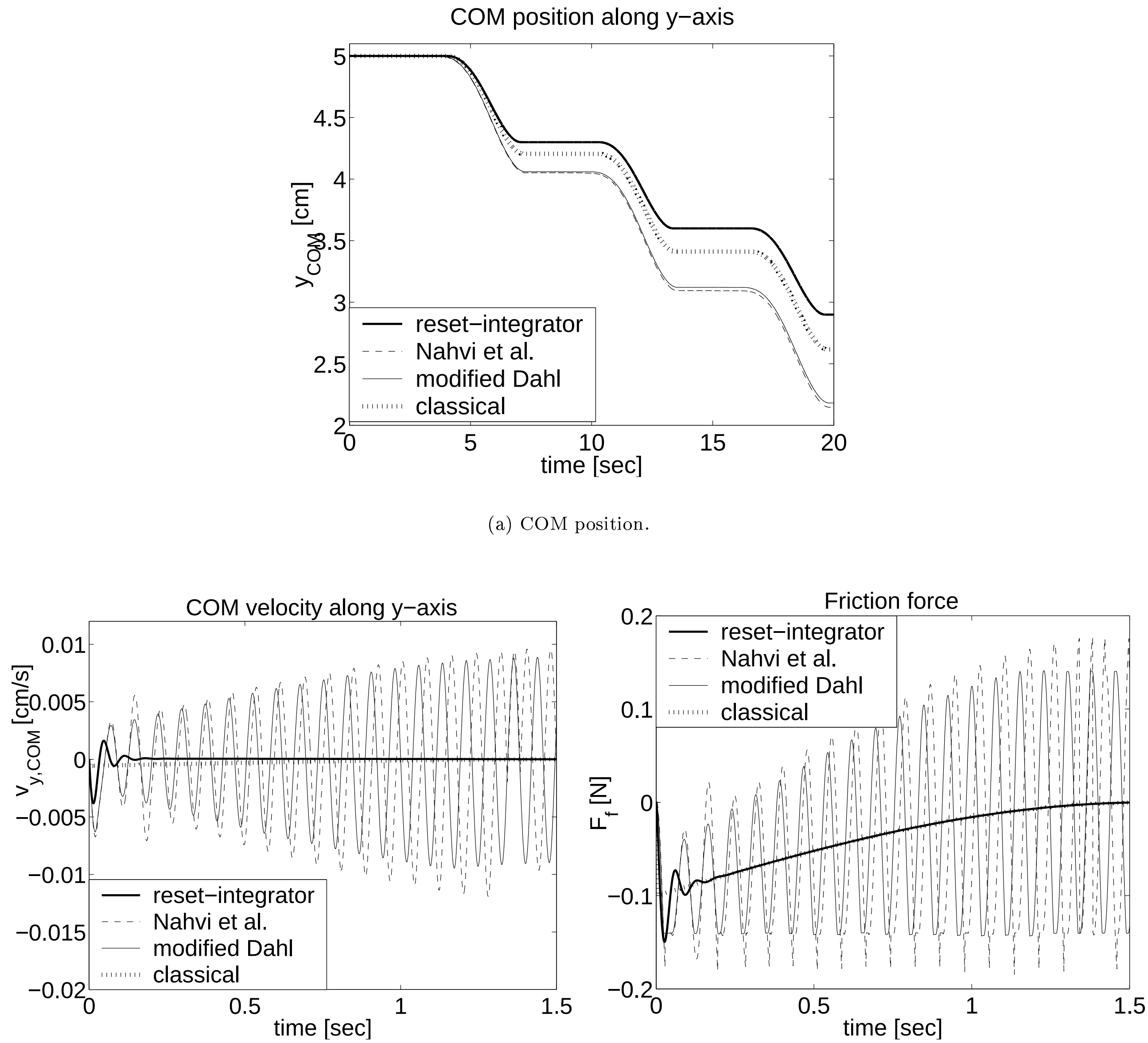


Figure 4.8: Classical approximation of Coulomb friction.

To experimentally validate the reset-integrator dry friction model, the same controlled interaction is implemented on the haptic simulation system presented in Chapter 3, using the synchronous communication paradigm. In the experiment, users insert the peg into the hole, release the device handle, and their hand is replaced by the unidirectional sinusoidal force $f_y = (-1 - sin(t))$N aligned with the hole axis. All investigated friction models are used for simulating contacts with friction. Their parameters are adjusted such that the contacts have coefficients of static and kinetic friction $\mu_s = 0.25$ and $\mu_k = 0.2$, respectively, a breakaway distance $z_0 = 0.01$mm, and a velocity dead band $v_{min} = 0.1$mm/s. The reset-integrator model is critically damped, ($\beta = 118$). The trajectory of the peg COM and the forces applied to users along the hole axis are plotted in Figure 4.10.

The experimental results confirm that all friction models exhibit slip and stick. However, some transitions from stick to slip are missed when dry friction is simulated using the model based on the human finger pad characteristics. The experiments also demonstrate that the driftless Dahl model and the model based on the human finger pad characteristics introduce perceivable compliance during stiction. This might not always be desirable for rendering frictional contact between rigid bodies. The critically damped reset-integrator model successfully eliminates such compliance. At the same time, the model can easily be adjusted to allow users to perceive oscillations during stiction by decreasing its damping coefficient β. Hence, the reset-integrator friction simulation technique can render a wider range of behaviors during stiction. The advantage of the reset-integrator model over the classical friction model is that it does not exhibit drift [61]. Presently, the reset-integrator model of dry friction is implemented by critically damping the oscillations during stiction. A physically meaningful choice of a damping coefficient requires future user studies.

The friction and contact models in Equations (4.5) and (4.6) show that only state information is used



(a) COM position.

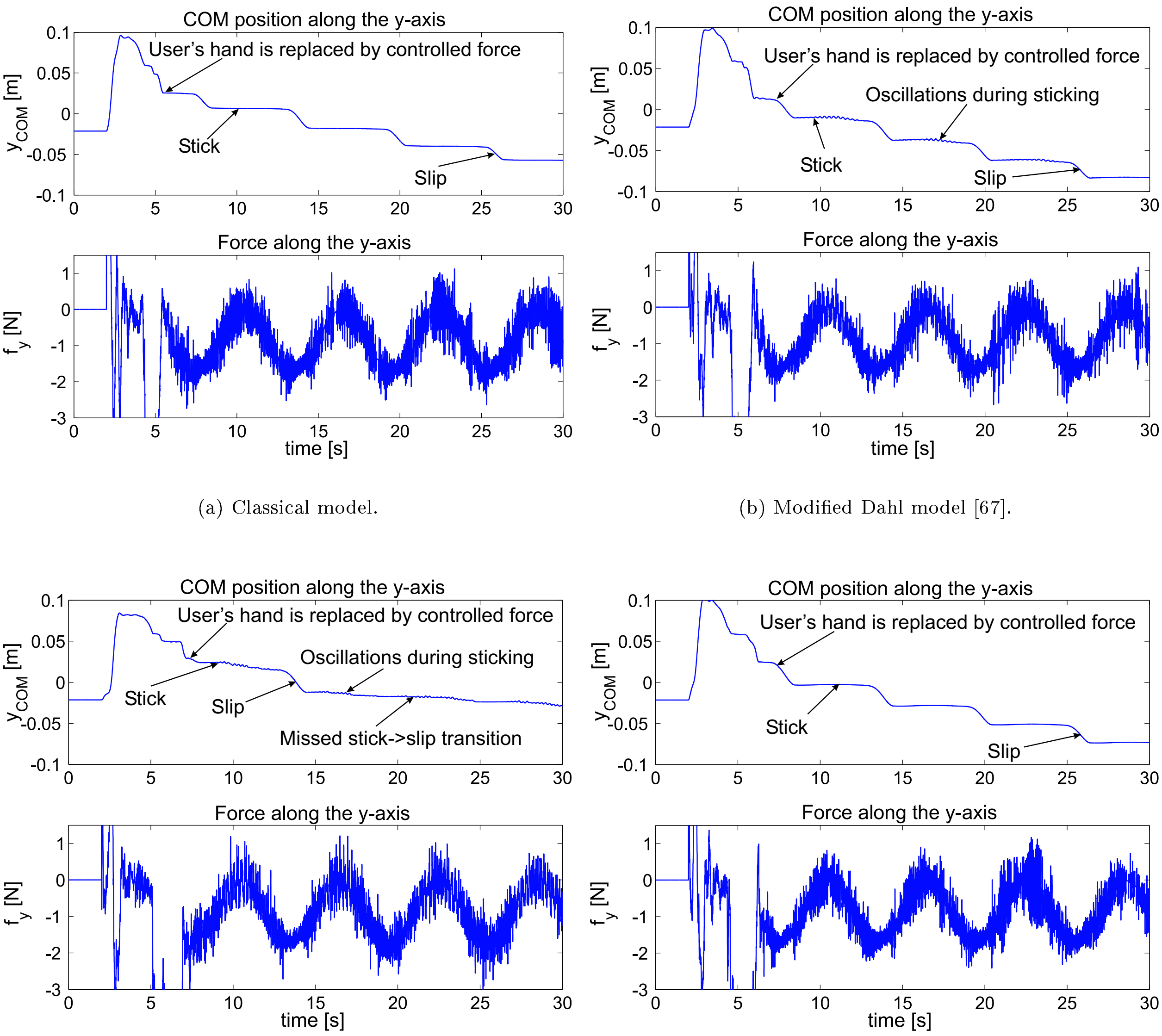
(b) Sliding velocity.

(c) Resultant friction force.

Figure 4.9: Peg-in-hole simulation. Friction is simulated using the critically damped reset-integrator model [61], the model proposed by Nahvi *et al.* [113], the modified Dahl model [67], and the classical dry friction approximation.

in the impulse-augmented penalty simulation to compute the contact forces in Equation (4.5). Therefore, Equation (4.1) can be directly solved for the configuration space acceleration:

$$\ddot{\mathbf{q}} = \mathbf{D}^{-1}\left(\sum_{i=1}^{c}\mathbf{J}_i^T\mathbf{f}_i + \mathbf{J}_h^T\mathbf{F}_h - \mathbf{B} - \mathbf{G}\right). \tag{4.13}$$



(a) Classical model.

(b) Modified Dahl model [67].

(c) Model based on human finger-pad characteristics [113].

(d) Reset-integrator model [61].

Figure 4.10: Forces felt by users and COM trajectories of a virtual peg sliding with friction in a tight hole due to a sinusoidal force applied to its COM. Various friction models are used to simulate dry friction. Note that all models render slip and stick. Perceivable oscillations occur during stiction when friction is simulated using the modified Dahl model [67] (Figure 4.10(b)) and the model based on human finger-pad characteristics [113] (Figure 4.10(c)). Some slip to stick transitions are missed by the model based on the human finger-pad characteristics [113] (Figure 4.10(c)).

In Equation (4.13), the dependence on the instantaneous state of all terms on the right hand side of the equation is implied. The configuration space acceleration is then integrated using a fixed step size integrator compatible with the requirements of the haptic control loop.

The dynamics of resting contact discussed in this section are typical for penalty-based simulations. They are used to approximately enforce rigidity during contact. Exact enforcement of rigidity upon contact is accomplished through implementing constraint-based colliding contact dynamics. These dynamics are presented in the following section.

4.4 Colliding contact

The colliding contact state is introduced in the impulse-augmented penalty approach to improve the penalty-based approximation of unyielding contacts. In conjunction with the fixed step size of the haptic simulation, this state requires multiple collisions to be resolved simultaneously. Rather than incorporating contact rigidity and dry friction into a complementarity formulation, the proposed approach uses three simplifying assumptions:

- that impulses develop at all contacts where bodies move into each other during a collision;
- that velocities of all colliding contact points obey Newton's restitution law;
- and, that collisions are frictionless.

Unlike more accurate multiple collision models based on complementarity formulations [6,63,112,120], this new technique is non-iterative and requires no further assumptions on the value of the coefficient of restitution or on the shape of colliding objects.

The colliding contact dynamics are obtained through time integration of Equation (4.1):

$$\mathbf{D}\dot{\mathbf{q}} = \mathbf{D}\dot{\mathbf{q}}_0 + \sum_{i=1}^{c} \int_{t_0}^{t} \mathbf{J}_i^T \mathbf{f}_i dt = \mathbf{D}\dot{\mathbf{q}}_0 + \sum_{i=1}^{c} \mathbf{J}_i^T \mathbf{p}_i. \qquad (4.14)$$

In Equation (4.14), $\mathbf{D}\dot{\mathbf{q}}_0$ and $\mathbf{D}\dot{\mathbf{q}}$ are the pre- and post-collision configuration space momenta and $\mathbf{p}_i = \int_{t_0}^{t} \mathbf{f}_i dt$ is the impulse at the i-th contact. Since collisions are modeled as instantaneous events, i.e., $t \to t_0$, the hand wrench and the gravitational forces do not contribute impulses to the impulse and momentum balance of the system. Furthermore, apart from the collision impulses, no other external impulses are applied to the contact group.

In addition, collisions are assumed frictionless:

$$\mathbf{p}_i = p_i \mathbf{n}_i. \tag{4.15}$$

In Equation (4.15), p_i is the magnitude of the i-th contact impulse. Then, the configuration space dynamics of colliding contacts become:

$$\mathbf{D}\dot{\mathbf{q}} = \mathbf{D}\dot{\mathbf{q}}_0 + \sum_{i=1}^{c} \mathbf{J}_i^T \mathbf{n}_i p_i = \mathbf{D}\dot{\mathbf{q}}_0 + \mathcal{J}_c^T \mathbf{p}, \tag{4.16}$$

where:

$$\mathbf{p} = \left(p_1 \dots p_i \dots p_c \right)^T \tag{4.17}$$

is the vector of contact impulses, and:

$$\mathcal{J}_c = \left[\mathbf{J}_1^T \mathbf{n}_1 \dots \mathbf{J}_i^T \mathbf{n}_i \dots \mathbf{J}_c^T \mathbf{n}_c \right]^T \tag{4.18}$$

is the contact Jacobian. For a contact group with d DOFs and c colliding contacts, Equation (4.16) represents a set of d equations with $d + c$ unknowns, the post-collision configuration space velocity $\dot{\mathbf{q}}$ and the contact impulses $\mathbf{p}$. The additional assumptions needed to solve this system are provided by the various collision laws proposed in the literature [112], [63], or [6, 120]. In this work, Newton's restitution rule is used, because it allows the development of a non-iterative solution that imposes no restrictions on the coefficient of restitution or the shape of the colliding objects. Moreover, the kinetic energy of the contact group does not increase during collisions.

For one colliding contact, Newton's restitution hypothesis relates the pre-collision (v_{n0}) and post-collision (v_n) normal contact velocities through the coefficient of restitution e:

$$v_n = -e v_{n0}. \tag{4.19}$$

The coefficient of restitution $e \in$ [0,1] describes the nature of the collision, with $e = 1$ corresponding to a perfectly elastic collision (no energy loss), and $e = 0$ corresponding to a perfectly plastic collision.

In configuration space, Equation (4.19) becomes:

$$\mathbf{n}^T \mathbf{J}_{b_i} \dot{\mathbf{q}} = -e \mathbf{n}^T \mathbf{J}_{b_i} \dot{\mathbf{q}}_0 \tag{4.20}$$

at a collision between body b_i of the contact group and a static environment, and:

$$\mathbf{n}^T \left(\mathbf{J}_{b_i} - \mathbf{J}_{b_j}\right) \dot{\mathbf{q}} = -e\mathbf{n}^T \left(\mathbf{J}_{b_i} - \mathbf{J}_{b_j}\right) \dot{\mathbf{q}}_0 \tag{4.21}$$

at a self-collision, i.e., a collision between bodies b_i and b_j of the contact group. A more restrictive condition is imposed at a self-collision in the proposed approach. Namely, the second simplifying assumption is imposed on the contact group by requiring it to obey:

$$\mathbf{n}^T \mathbf{J}_{b_i} \dot{\mathbf{q}} = -e\mathbf{n}^T \mathbf{J}_{b_i} \dot{\mathbf{q}}_0 \tag{4.22}$$

and:

$$\mathbf{n}^T \mathbf{J}_{b_j} \dot{\mathbf{q}} = -e\mathbf{n}^T \mathbf{J}_{b_j} \dot{\mathbf{q}}_0 \tag{4.23}$$

simultaneously. Equations (4.22) and (4.23) ensure both that Newton's restitution law is observed and that the proposed collision resolution technique maintains system passivity, as shown in subsequent derivations. Using Equations (4.20), (4.22), and (4.23), Newton's restitution law is restated as:

$$\mathcal{J}_c \dot{\mathbf{q}} = -e\mathcal{J}_c \dot{\mathbf{q}}_0. \tag{4.24}$$

Equation (4.24) represents a set of c equations, where self-collisions are counted once on each colliding body. Note also that the equal sign in Equation (4.24) embeds the first simplifying assumption used for resolving collisions.

In the impulse-augmented penalty simulation approach, Equations (4.16) and (4.24) describe the dynamics of colliding contact of a contact group with d DOFs and c simultaneous collisions. Their resolution, theorems for system passivity for both independent and overdetermined constraints, and the computation of the configuration space impulsive interactions applied to users are presented in the following subsection.

4.4.1 Passive collision resolution

This section starts by showing that a contact group with one frictionless contact is passive if its colliding contact dynamics are resolved using Newton's hypothesis. The result is then extended to a contact group with multiple independent and overdetermined contacts for which Equation (4.24) is used to ensure that Newton's collision law is obeyed.

4.4.1.1 Independent constraints

Passivity of a contact group with a single colliding contact is shown by proving that:

Theorem 1 *If a contact group described by the momentum equation:*

$$\mathbf{D}\dot{\mathbf{q}} = \mathbf{D}\dot{\mathbf{q}}_0 + \mathcal{J}_c^T \mathbf{p}$$

has one frictionless colliding contact and the post-collision normal contact velocity is given by:

$$v_n = -e v_{n0},$$

where $e \in [0,1]$ is the coefficient of restitution, then the post-collision kinetic energy of the contact group is:

$$KE = KE_0 - \left(1 - e^2\right) \dot{\mathbf{q}}_0^T \mathcal{J}_c^T \overline{\mathcal{J}}_c^T \mathbf{D} \overline{\mathcal{J}}_c \mathcal{J}_c \dot{\mathbf{q}}_0 \leq KE_0 \tag{4.27}$$

where $\overline{\mathcal{J}}_c = \mathbf{D}^{-1}\mathcal{J}_c^T \left(\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T\right)^{-1}$ is the dynamically consistent inverse of $\mathcal{J}_c$ [81].

Proof The proof starts by computing the contact impulse by substitution of Equation (4.24) into Equation (4.16):

$$-e\mathcal{J}_c\dot{\mathbf{q}}_0 = \mathcal{J}_c\dot{\mathbf{q}}_0 + \mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\mathbf{p} \quad \Rightarrow \quad \mathbf{p} = -\left(1+e\right)\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c\dot{\mathbf{q}}_0. \tag{4.28}$$

Then, the post-collision configuration space velocity results after substitution from Equation (4.28) in Equation (4.16):

$$\dot{\mathbf{q}} = \dot{\mathbf{q}}_0 - \left(1+e\right)\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c\dot{\mathbf{q}}_0 = \left(\mathbf{I} - \left(1+e\right)\overline{\mathcal{J}}_c\mathcal{J}_c\right)\dot{\mathbf{q}}_0 \tag{4.29}$$

where $\mathbf{I}$ is the $d \times d$ identity matrix.

Next it is shown that:

$$\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c - \mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c = 0. \tag{4.30}$$

Indeed, using the definition of the dynamically consistent inverse of $\mathcal{J}_c$ and the symmetry of the inertia matrix:

$$\begin{aligned}
& \mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c - \mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c = \\
= \; & \mathbf{D}\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c - \mathcal{J}_c^T\left(\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\right)^T\mathbf{D}\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c = \\
= \; & \mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c - \mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-T}\mathcal{J}_c\mathbf{D}^{-T}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c = \\
= \; & \mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c - \mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}\mathcal{J}_c = 0.
\end{aligned} \tag{4.31}$$

In Equation (4.31), $\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-T} = \left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}$ and $\mathbf{D}^{-T} = \mathbf{D}^{-1}$ since they are symmetric. Furthermore:

$$\left(\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c - \mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c\right)^T = \mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D} - \mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c = 0. \tag{4.32}$$

Then, the post-collision kinetic energy of the system is be computed as follows:

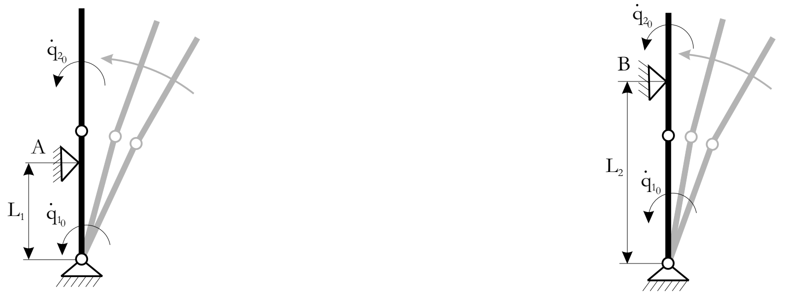
$$\begin{aligned}
KE \; = \; & \frac{1}{2}\dot{\mathbf{q}}^T\mathbf{D}\dot{\mathbf{q}} = \frac{1}{2}\dot{\mathbf{q}}_0^T\left(\mathbf{I} - (1+e)\,\mathcal{J}_c^T\overline{\mathcal{J}}_c^T\right)\mathbf{D}\left(\mathbf{I} - (1+e)\,\overline{\mathcal{J}}_c\mathcal{J}_c\right)\dot{\mathbf{q}}_0 = \\
= \; & \frac{1}{2}\dot{\mathbf{q}}_0^T\mathbf{D}\dot{\mathbf{q}}_0 - (1+e)\,\frac{1}{2}\dot{\mathbf{q}}_0^T\left(\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c + \mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D} - (1+e)\,\mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c\right)\dot{\mathbf{q}}_0 = \\
= \; & KE_0 - (1+e)\,\frac{1}{2}\dot{\mathbf{q}}_0^T\left(2\mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c - (1+e)\,\mathcal{J}_c^T\overline{\mathcal{J}}_cT\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c\right)\dot{\mathbf{q}}_0 = \\
= \; & KE_0 - \left(1 - e^2\right)\dot{\mathbf{q}}_0^T\mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c\dot{\mathbf{q}}_0.
\end{aligned} \tag{4.33}$$

Since $\mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c$ is symmetric, it is positive semi-definite and:

$$KE \leq KE_0 \quad \forall e \in [0,1] \quad \text{Q.E.D.} \tag{4.34}$$

The contact impulse due to one collision given in Equation (4.28) is equal to the contact impulse computed in prior complementarity formulations [112]. Hence, the simplifying assumptions embedded in (4.24) involve no further approximation for the case of a single frictionless colliding contact. Moreover, the proposed collision resolution method uses the dynamically consistent inverse of the collision Jacobian, i.e., it is coordinate invariant.

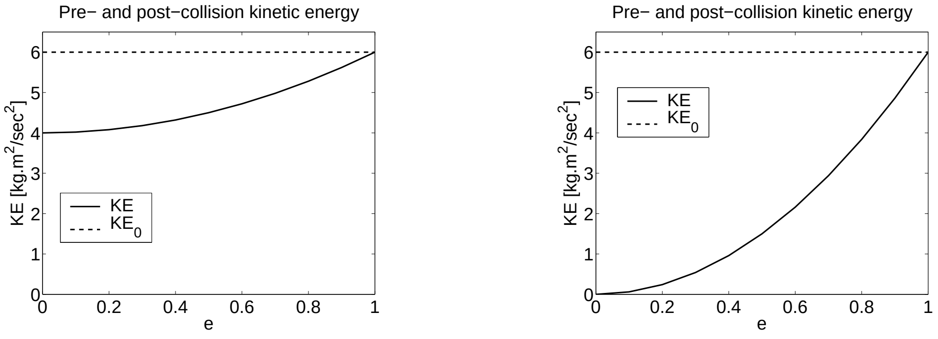
Kinetic energy is conserved during a perfectly elastic collision ($e = 1$). The loss of kinetic energy during a plastic collision ($e < 1$) depends both on contact properties, as given by the coefficient of restitution e, and on contact geometry and the contact group topology and geometry, embedded in $\mathcal{J}_c$. Various contact group topologies and geometries and various contact geometries result in either a total or a partial loss of kinetic energy during a plastic collision. This is illustrated in Figure 4.12 for two different contact geometries of the two link planar manipulator depicted in Figure 4.11[3]. Note that the manipulator loses all kinetic energy during a perfectly plastic collision between the distal link and a static environment, and it loses only a part of its kinetic energy during a perfectly plastic collision between the base link and a static environment.



(a) Base link colliding with the environment.

(b) Distal link colliding with the environment.

Figure 4.11: Two link planar manipulator whose loss of kinetic energy for various values of the coefficient of restitution is shown in Figure 4.12 for two contact geometries. The simplifying assumption in Equation (4.24) imposes the particular solution shown in Figure 4.12(b).



(a) Base link colliding with the environment ($L_1 = 0.7$m).

(b) Distal link colliding with the environment ($L_2 = 1.5$m).

Figure 4.12: Loss of kinetic energy of the planar two link manipulator in Figure 4.11 during one frictionless collision for two contact geometries.

[3] The manipulator has link lengths $l_1 = l_2 = 1$m, link masses $m_1 = m_2 = 1$kg, configuration space position $\mathbf{q} = \begin{pmatrix} \frac{\pi}{2} & 0 \end{pmatrix}^T$rad, and pre-collision configuration space velocity $\dot{\mathbf{q}}_0 = (1 \quad 1)^T$rad/s. The environment constraints at A and B are such that $L_1 < 1$m and 1m$< L_2 < 2$m, respectively.

The collision resolution method employed for one colliding contact can be directly applied to resolve multiple collisions simultaneously if the contact constraints are independent, i.e., $\mathcal{J}_c$ is full row rank, and Equation (4.24) is imposed to ensure that the post-collision configuration space velocity obeys Newton's hypothesis. Then, $\mathcal{J}_c$ is full row rank and the matrix $\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T$ is invertible. This can be easily shown by considering that its singular values of bmD^{-1} and $\mathcal{J}_c \mathcal{J}_c^T$, $\Sigma_{min}\left(\mathbf{D}^{-1}\right)$ and $\Sigma_{min}\left(\mathcal{J}_c \mathcal{J}_c^T\right)$, respectively, obey:

$$\dot{\mathbf{q}}^T \mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T \dot{\mathbf{q}} \geq \Sigma_{min}\left(\mathbf{D}^{-1}\right) \dot{\mathbf{q}}^T \mathcal{J}_c \mathcal{J}_c^T \dot{\mathbf{q}} \geq \Sigma_{min}\left(\mathbf{D}^{-1}\right) \Sigma_{min}\left(\mathcal{J}_c \mathcal{J}_c^T\right) \dot{\mathbf{q}} \dot{\mathbf{q}}^T > 0 \quad \forall \dot{\mathbf{q}} \neq \mathbf{0}, \tag{4.35}$$

where $\Sigma_{min}\left(\mathbf{D}^{-1}\right)$ is the minimum singular value of $\mathbf{D}^{-1}$. Equation (4.35) proves that $\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T$ is positive definite, hence invertible.

4.4.1.2 Overdetermined constraints

If the contact constraints are overdetermined, $\mathcal{J}_c$ is rank deficient and the matrix $\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T$ is not invertible. Nevertheless, its pseudo-inverse $\left(\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T\right)^{\dagger}$ can be used to compute the contact impulses $\mathbf{p}$, and the post-collision configuration space velocity $\dot{\mathbf{q}}$ according to:

$$\begin{aligned} \mathbf{p} &= -(1+e)\left(\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T\right)^{\dagger} \mathcal{J}_c \dot{\mathbf{q}}_0 & (4.36) \\ \dot{\mathbf{q}} &= \dot{\mathbf{q}}_0 - (1+e) \mathbf{D}^{-1} \mathcal{J}_c^T \left(\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T\right)^{\dagger} \mathcal{J}_c \dot{\mathbf{q}}_0. & (4.37) \end{aligned}$$

Passivity of a contact group with overdetermined colliding constraints resolved simultaneously according to Equation (4.37) results from the following theorem:

Theorem 2 *If a contact group described by the momentum equation:*

$$\mathbf{D}\dot{\mathbf{q}} = \mathbf{D}\dot{\mathbf{q}}_0 + \mathcal{J}_c^T \mathbf{p}, \tag{4.38}$$

has c overdetermined frictionless colliding contacts and its post-collision configuration space velocity is given by:

$$\mathcal{J}_c \dot{\mathbf{q}} = -e \mathcal{J}_c \dot{\mathbf{q}}_0, \tag{4.39}$$

where $e \in [0,1]$ *is the coefficient of restitution, then the post-collision kinetic energy of the contact group is:*

$$KE = KE_0 - \left(1-e^2\right) \dot{\mathbf{q}}_0^T \mathcal{J}_n^T \overline{\mathcal{J}}_n^T \mathbf{D} \overline{\mathcal{J}}_n \mathcal{J}_n \dot{\mathbf{q}}_0 \leq KE_0. \tag{4.40}$$

In Equation (4.40), $\overline{\mathcal{J}}_n = \mathbf{D}^{-1} \mathcal{J}_n^T \left(\mathcal{J}_n \mathbf{D}^{-1} \mathcal{J}_n^T\right)^{-1}$ *is the dynamically consistent inverse of* $\mathcal{J}_n$*, and* $\mathcal{J}_c^T =$

$\begin{bmatrix}\mathcal{J}_n^T & \mathcal{J}_r^T\end{bmatrix}^T$*, with* $\mathcal{J}_n$ *full row rank.*

Proof The proof follows the same reasoning as the proof of Theorem 1, where $\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{-1}$ is replaced by $\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}$. The proof holds since:

$$\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger} = \left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}. \tag{4.41}$$

Furthermore, it is shown in Appendix A that:

$$\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}\mathcal{J}_c = \mathcal{J}_n^T\left(\mathcal{J}_n\mathbf{D}^{-1}\mathcal{J}_n^T\right)^{-1}\mathcal{J}_n, \tag{4.42}$$

where $\mathcal{J}_c^T = [\mathcal{J}_n^T \quad \mathcal{J}_r^T]^T$ and $\mathcal{J}_n$ is full row rank, i.e., $rank(\mathcal{J}_n) = rank(\mathcal{J}_c) = n$. Then:

$$\begin{aligned}
KE &= KE_0 - \left(1-e^2\right)\dot{\mathbf{q}}_0^T\mathcal{J}_c^T\overline{\mathcal{J}}_c^T\mathbf{D}\overline{\mathcal{J}}_c\mathcal{J}_c\dot{\mathbf{q}}_0 = \\
&= KE_0 - \left(1-e^2\right)\dot{\mathbf{q}}_0^T\mathcal{J}_c^T\left(\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}\right)^T\mathbf{D}\left(\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}\right)\mathcal{J}_c\dot{\mathbf{q}}_0 = \\
&= KE_0 - \left(1-e^2\right)\dot{\mathbf{q}}_0^T\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}\mathcal{J}_c\dot{\mathbf{q}}_0 = \\
&= KE_0 - \left(1-e^2\right)\dot{\mathbf{q}}_0^T\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}\mathcal{J}_c\dot{\mathbf{q}}_0 = \\
&= KE_0 - \left(1-e^2\right)\dot{\mathbf{q}}_0^T\mathcal{J}_n^T\overline{\mathcal{J}}_n^T\mathbf{D}\overline{\mathcal{J}}_n\mathcal{J}_n\dot{\mathbf{q}}_0.
\end{aligned} \tag{4.43}$$

Since $\mathcal{J}_n^T\overline{\mathcal{J}}_n^T\mathbf{D}\overline{\mathcal{J}}_n\mathcal{J}_n$ is positive semi-definite and $e \in [0,1]$, it follows that the kinetic energy of the contact group does not increase during simultaneous collision resolution when the constraints are overdetermined regardless of the value of the coefficient of restitution, i.e. $KE \leq KE_0 \quad \forall e \in [0,1]$ Q.E.D.

Similar to the case of independent constraints, the loss of kinetic energy depends both on contact properties, through the coefficient of restitution e, and on the topology and geometry of the contact group and the geometry of contact, through $\mathcal{J}_n$. Kinetic energy is conserved during perfectly elastic collisions ($e = 1$).

From Equation (4.42), it also follows that:

$$\begin{aligned}
\dot{\mathbf{q}} &= \dot{\mathbf{q}}_0 - (1+e)\,\mathbf{D}^{-1}\mathcal{J}_c^T\left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^{\dagger}\mathcal{J}_c\dot{\mathbf{q}}_0 = \\
&= \dot{\mathbf{q}}_0 - (1+e)\,\mathbf{D}^{-1}\mathcal{J}_n^T\left(\mathcal{J}_n\mathbf{D}^{-1}\mathcal{J}_n^T\right)^{-1}\mathcal{J}_n\dot{\mathbf{q}}_0.
\end{aligned} \tag{4.44}$$

In other words, the post-collision state of the contact group is the same regardless of whether the collisions are resolved using the pseudo-inverse technique or the constraint overdeterminancy is eliminated before collision resolution. Hence, the pseudo-inverse method is equivalent to selecting a set of independent constraints and simultaneously resolving the collisions at these contacts as described in Section 4.4.1.1.

4.4.2 Rendering collisions to users

Collision impulses are rendered to users as impulsive forces through the haptic controller shown in Figure 4.1 at the beginning of this chapter. The configuration space impulsive torques to be applied to users' hand $\mathbf{T}_{env}$ are computed such that, when integrated over one time step of the haptic simulation, they induce the same change in the configuration space momentum of the contact group as the simulated collision impulses:

$$\mathbf{T}_{env} = \frac{\mathcal{J}_c^T \mathbf{p}}{\Delta t}. \tag{4.45}$$

In Equation (4.45), Δt is the time step of the haptic simulation and $\mathbf{p}$ are the contact impulses, computed according to Equations (4.28) and (4.36).

The key feature of the haptic interaction within the impulse-augmented penalty virtual world described above is that it enables users to feel collisions when new contacts arise. This is in contrast to existing haptic interaction paradigms which apply only penalty-like forces to users. To demonstrate the advantage of the proposed approach, the next section compares it to existing methods, as well as to the "ideal" case (i.e., physical interaction) through simulations and experiments.

4.5 The performance of the impulse-augmented penalty simulation

In this section, the performance of the impulse-augmented simulation approach is evaluated against the performance of the penalty-based and the constraint-based approaches through simulated and experimental user interactions within a planar virtual environment[4]. The evaluation is initially carried out assuming that the

[4] Note that the dynamics in Sections 4.3 and 4.4 are suitable for rigid body interaction within spatial virtual environments. However, the approach is validated only for rigid body interaction within a planar virtual world in this thesis. This is due to the fact that only planar (f_x, f_y, τ_z) and point (f_x, f_y, f_z) haptic interaction systems are available in the Robotics and Control Laboratory. Furthermore, full rigid body force and torque feedback is required (f_x, f_y, τ_z during planar interaction and f_x, f_y, f_z, τ_x, τ_y, τ_z during spatial interaction) to guarantee that the kinetic energy of the user's hand will not increase during collisions as a result of the impulsive feedback. Unstable interaction may arise due to the lack of torque feedback.

haptic interface can fully apply the simulated impulses to users. Device limitations are then taken into account and solutions are designed to address them.

Figure 4.13 depicts the virtual environment employed in the simulations. Users manipulate a rectangular peg by applying forces and torques at its center of mass. The virtual world is connected to the haptic interface through a unilateral coupler [155], [127] when generated using a constraint-based approach, and through a four-channel teleoperation controller [136] when generated using the impulse-augmented penalty and the penalty-based approaches. The unilateral coupler extends to rigid body interaction the controller used in the god-object [155] and the virtual proxy [127] point interaction techniques[5]. It implements two spring-damper connections (one translational and one rotational) between the virtual tool and the device during constrained motion and is inactive during free motion. The teleoperation controller coordinates both forces and positions between the virtual tool and the haptic interface, as explained in Section 4.1. Schematics of the mechanical equivalents of both controllers for 1 DOF interaction are shown in Figure 4.14.

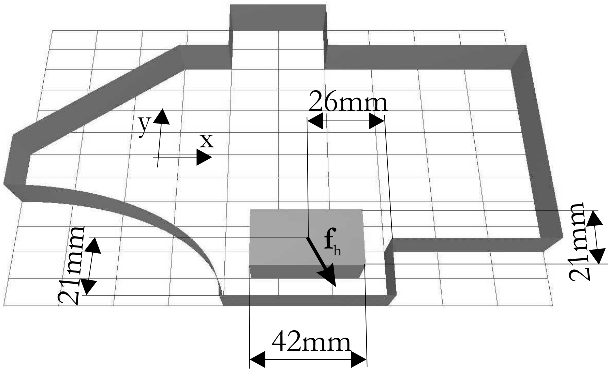
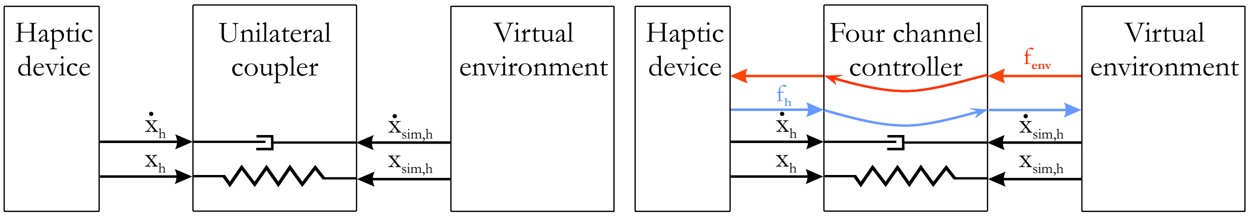


Figure 4.13: Planar virtual world used in simulations and experiments. Starting from rest and the position shown, the rectangular object is pushed into the lower right corner by a controlled constant force.

In the simulations, the user applies a wrench $\mathbf{F}_h = (0.32\text{N} \quad -0.4\text{N} \quad 0\text{Nm})^T$ on the peg, i.e., they push the peg towards the lower right corner of the virtual enclosure. The user-applied wrench is such that the peg hits the vertical wall first. It then slides along this wall until it hits the horizontal wall and stops. This simple interaction is chosen because it illustrates the performance of the approach for overdetermined constraints. The peg has dimensions $l_1 = 0.042\text{m}$ and $l_2 = 0.021\text{m}$, mass $m = 2\text{kg}$, and moment of inertia $I = 0.005\text{kgm}^2$. The virtual walls have stiffness $K_{wall} = 4000\text{N/m}$ and damping $B_{wall} = 30\text{N/(m/s)}$. Collisions are considered perfectly plastic ($e = 0$). The stiffness and damping of the position coordination channels of the teleoperation controller connecting the proposed and the penalty-based virtual environments and of the unilateral coupler

[5] However, since it is not clear how the god-object and the virtual proxy simulations can be extended to rigid body interaction, the virtual world is evolved using forward dynamics algorithms similar to those developed in graphics [11].



(a) The unilateral coupler during constrained motion.

(b) The four channel controller.

Figure 4.14: 1 DOF mechanical equivalents of the controllers connecting the haptic device and the virtual environment. Controller parameters are given in Table 4.1.

connecting the constraint-based virtual world to the device are given in Table 4.1. The parameters of the teleoperation controller are optimized for transparency [136], while those of the unilateral coupler are chosen to match the impedance of the virtual contacts. To match the planar interface used for experiments, the haptic device is modeled as an impedance device. Furthermore, the time step of the simulation and of the controllers connecting the haptic interface and the virtual environment is chosen equal to $\frac{1}{512}$s. In addition, the haptic device is considered to have purely inertial dynamics and to be kinematically equivalent to the proxy, i.e., the local device controller is not modeled. The user's hand is modeled as a pure force source, which is a worst-case scenario for stability when an impedance device is used [64].

Table 4.1: Parameters of the teleoperation controller and the unilateral coupler connecting the virtual environment to the haptic device.

Four channel Controller	**Unilateral coupler**
$\mathbf{K}_{coord} = \begin{pmatrix} 100\text{N/m} & 100\text{N/m} & 0.5\text{N/rad} \end{pmatrix}^T$	$\mathbf{K}_{cpl} = \begin{pmatrix} 1000\text{N/m} & 1000\text{N/m} & 2.5\text{N/rad} \end{pmatrix}^T$
$\mathbf{B}_{coord} = \begin{pmatrix} 70\text{N/(m/s)} & 70\text{N/(m/s)} & 0.375\text{N/(rad/s)} \end{pmatrix}^T$	$\mathbf{B}_{cpl} = \begin{pmatrix} 50\text{N/(m/s)} & 50\text{N/(m/s)} & 0.125\text{N/(rad/s)} \end{pmatrix}^T$

The performance of the haptic interaction paradigm proposed in this thesis is compared to that of existing techniques in the following two sections.

4.5.1 Best performance

In this section, the planar interaction described above is simulated by assuming that the haptic interface can fully apply the virtual impulses to the user. Hence, the best performance potentially achievable during haptic manipulation of impulse-augmented penalty virtual worlds is investigated. Interface limitations are considered and addressed in the following section.

Figures 4.15, 4.16, and 4.17 depict the users' hand trajectories, the forces felt by users, and the kinetic energy of the users' hand, respectively, while users move towards the virtual rigid corner, contact the vertical wall, and slide down along it until they contact the horizontal wall. In these figures, the various interaction paradigms are identified as follows. "Ideal", "IAPB", and "PB" represent interaction with a constraint-based, an impulse-augmented penalty-based, and a penalty-based virtual environment, respectively, all rendered to users through the four channel controller shown in Figure 4.14(b). "CB$_{uc}$" represents interaction with a constraint-based virtual world rendered to users through the unilateral coupler shown in Figure 4.14(a). Note that only the PB interaction has been demonstrated in existing haptics research. Note also that the "ideal" trajectory represents the peg interaction with a perfectly rigid (i.e., real) environment. Therefore, this trajectory separates the free motion of the peg from its constrained motion. Positions of the COM of the peg to the left and above the lowest point on this trajectory represent free motion of the peg (i.e., peg not in contact with the environment). Positions of the COM of the peg to the right and below the lowest point on this trajectory correspond to peg penetration into the vertical and the horizontal wall, respectively (i.e., peg in contact with the vertical and the horizontal wall, respectively).

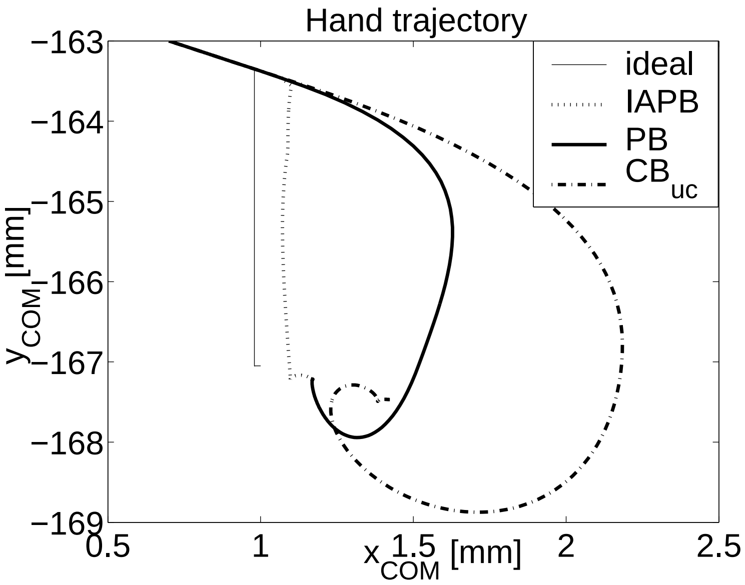
Figure 4.15 demonstrates that users' penetration into the virtual walls is smaller when users interact with a world generated using the impulse-augmented penalty approach than when they interact with a penalty-based or with a constraint-based virtual environment. This means that the trajectory of the user's hand is closer to the ideal trajectory, i.e., the interaction is more realistic, when the virtual world is simulated using the proposed approach. The user-perceived forces are closer to the ideal forces, too. Users feel large wrenches when new contacts arise (see Figure 4.16), and they feel wrenches that only balance the small hand wrench $\mathbf{F}_h = (0.32\text{N} \quad -0.4\text{N} \quad 0\text{Nm})^T$ during contact. Figure 4.17 demonstrates that users lose more kinetic energy upon contact when they interact with an impulse-augmented virtual world than when they interact with a penalty-based or with a constraint-based virtual environment. This means that the impulse-augmented penalty contacts improve the passivity of the virtual world compared to the other simulation paradigms, thereby improving the stability of the haptic interaction.

Note that, while the large impulsive forces improve the perceived rigidity of the virtual contacts, they may exceed the force capabilities of the device. Constraints imposed by the haptic device on the virtual manipulation of an impulse-augmented penalty simulation are considered in the following section.

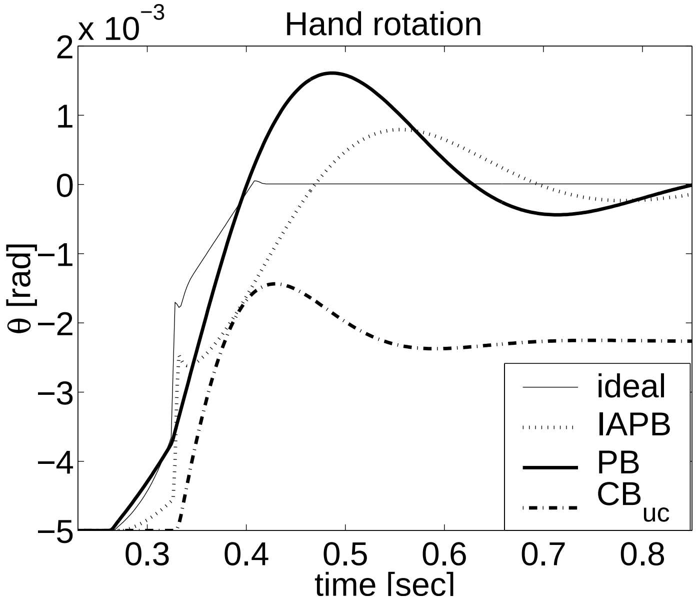
4.5.2 Accounting for device limitations

Several techniques can be used to account for device limitations during haptic interaction with impulse-augmented penalty virtual environments:

- Collision impulses can be saturated on the device. When this strategy is used, the simulation computes collision impulses according to Equation (4.36) and sends them to the four channel controller according to Equation (4.45). The controller then saturates the impulses to the maximum value that the actuators can apply to the device. Hence, full collision impulses are applied to the virtual tool in the simulation and saturated impulses are applied to the device. As a result, different amounts of kinetic energy are extracted from the virtual tool and the device during collision and the kinematic correspondence between the two can be significantly changed, depending on the system dynamics. Post-collision kinematic correspondence is re-established through the position coordination channel of the four channel controller. This channel is more compliant than the contact, and thus larger user violations of constraints can result.

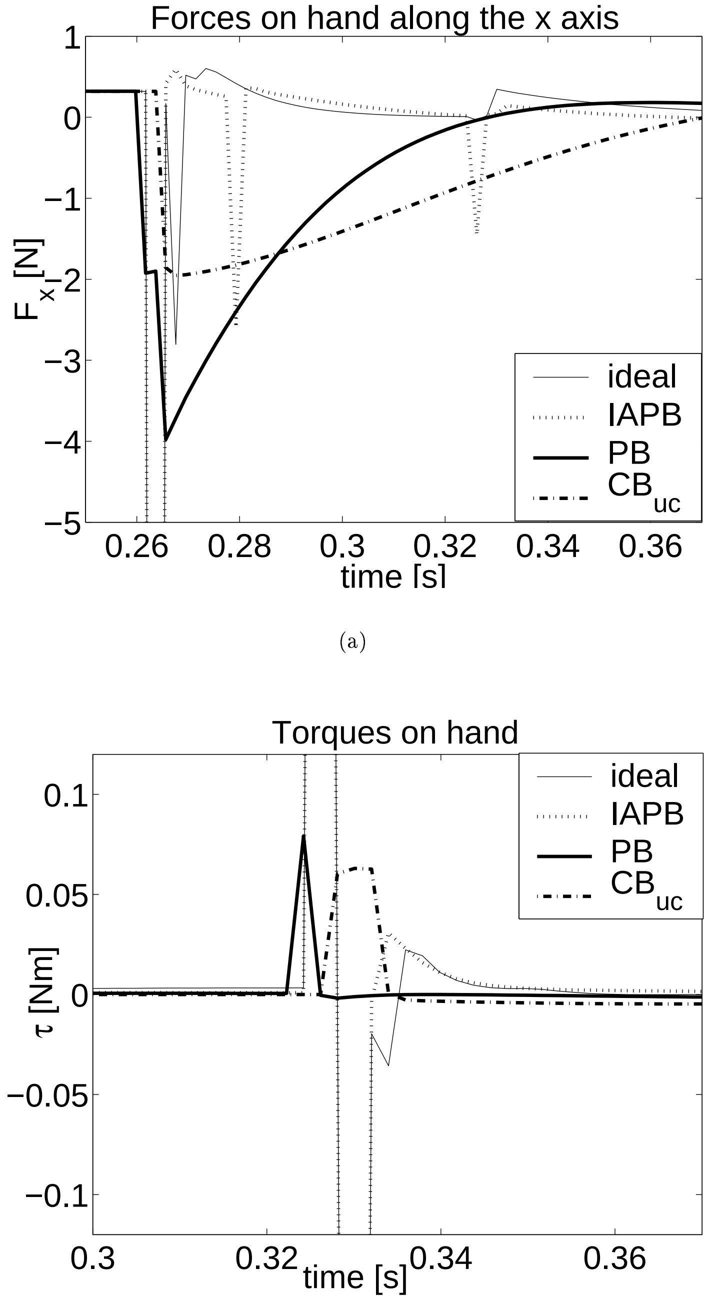


(a)



(b)

Figure 4.15: Simulated hand trajectories obtained when constraint-based ("ideal"), impulse-augmented penalty-based ("IAPB"), and penalty-based ("PB") interactions are applied to users by the four channel controller, and when constraint-based interactions are applied by the unilateral coupler ("CB_{uc}"). The device applies the simulated impulses to users in one step.



(a)

(b)

Figure 4.16: Simulated forces along the x-axis and torques applied to users by the four channel controller when the virtual world is generated using the constraint-based ("ideal"), the impulse-augmented penalty-based ("IAPB"), and the penalty-based ("PB") methods, and applied by the unilateral coupler when the virtual environment is generated using the constraint-based method ("CB_{uc}"). The device applies the simulated impulses to users in one step.

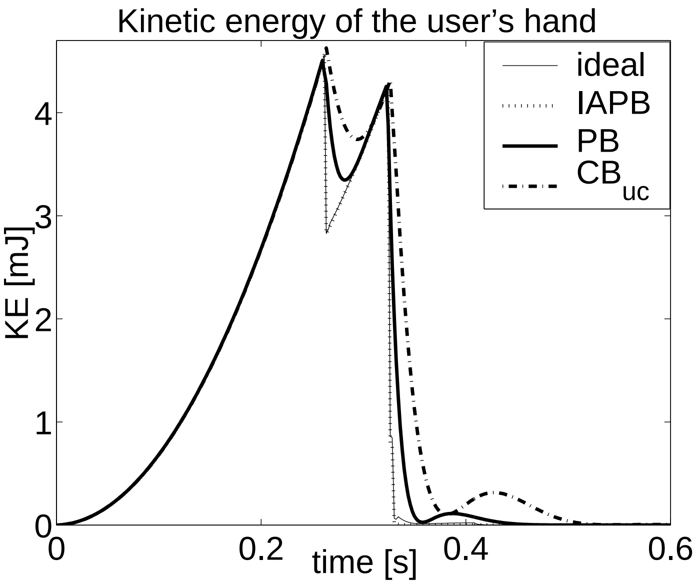


Figure 4.17: Kinetic energy of the users' hand when constraint-based ("ideal"), impulse-augmented penalty-based ("IAPB"), and penalty-based ("PB") interactions are appllied to users by the four channel controller, and when constraint-based interactions are applied by the unilateral coupler ("CB$_{uc}$"). The device applies the simulated impulses to users in one step.

- Collision impulses can be scaled in the simulation. When this strategy is used, the simulation computes collision impulses according to Equation (4.36) and scales them to the maximum value that the device actuators can apply on users. The scaled impulses are applied to the virtual tool and sent to the four channel controller. As a result, the same amount of kinetic energy is extracted from the virtual tool and the device during collision and their kinematic correspondence is maintained. However, the dissipated kinetic energy is less than that prescribed by the coefficient of restitution and the contact geometry. The simulation is altered and larger user penetrations into constraints occur.

- Collision impulses can be spread over several steps of the simulation. When this strategy is used, the simulation scales the collision impulses as explained above. However, the colliding contact group does not transition to resting contact if scaling is necessary. Rather, it transitions back to colliding contact and new collision impulses are computed at each step of the simulation until the force levels return to the range of the haptic device. As a result, the amount of kinetic energy extracted from the virtual tool and the device at each simulation step is the maximum allowable by the actuators, while the amount extracted over several steps is equal to that prescribed by coefficient of restitution and the contact geometry in Equation (4.40). Furthermore, the kinematic correspondence between the virtual tool and the device is preserved.

Simulated hand trajectories obtained by using these techniques (assuming that the maximum wrench capability of the device is $\mathbf{F}_{limit} = (15\text{N} \quad 15\text{N} \quad 1\text{Nm})^T$) are presented in Figure 4.18. In this figure, "p_{full}"

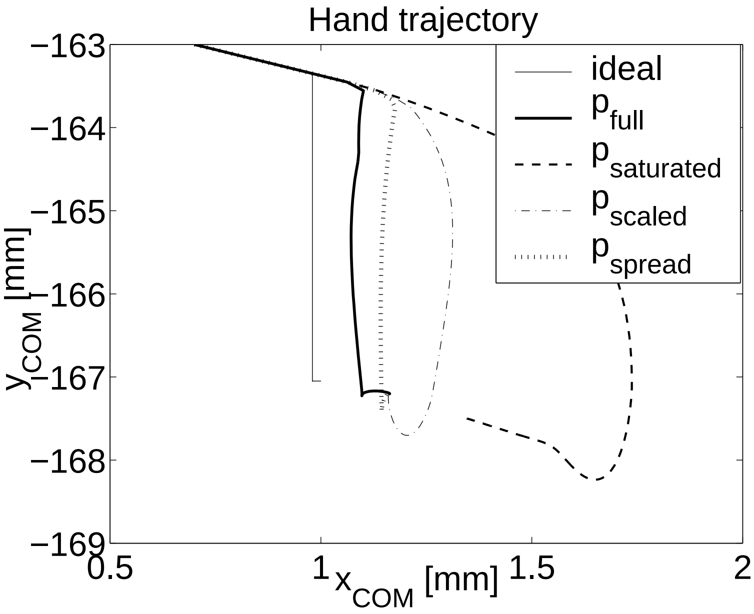
is the trajectory obtained when the device can fully apply the simulated collision impulses, "$\mathrm{p}_{saturated}$" is obtained when collision impulses are saturated on the device, "p_{scaled}" is obtained when collision impulses are scaled in the virtual environment, and "p_{spread}" is obtained when collision impulses are spread over several steps of the simulation. Moreover, "ideal" is the trajectory obtained when the device fully applies the impulses generated by a constraint-based simulation. The hand trajectory degrades as the wrench capabilities of the device decrease. The loss of performance is highest if the interaction forces are saturated on the device. In this case, full collision impulses are applied to the virtual tool which stops abruptly and only limited wrenches are applied to the user's hand, which continues to move. After collision, the user's hand is coordinated with the virtual tool through the position coordination channels of the haptic controller, whose stiffness $\mathbf{K}_{coord}$ and damping $\mathbf{B}_{coord}$ are much lower than those of the virtual walls, K_{wall} and B_{wall}. Hence, constraint violation is largest and settling time increases (transient response is poorest). The loss of performance due to limited wrench capabilities of the haptic interface is diminished most by spreading the collision impulses over several steps of the simulation. Therefore, this technique is adopted to overcome device force limitations during haptic manipulation of impulse-augmented penalty virtual worlds.

Both scaling and spreading of collision impulses are equivalent to adapting the coefficient of restitution to the device capabilities. As a result of this adaptation, the effective coefficient of restitution may be negative and the post-collision normal contact velocities may be negative (i.e., bodies may move into each other after collision resolution). Nevertheless, the energetic passivity of the proposed collision resolution approach is not affected by an adaptive coefficient of restitution[6]. Hence, the coupled stability of the haptic interaction system is unaffected by the adaptation of the coefficient of restitution.

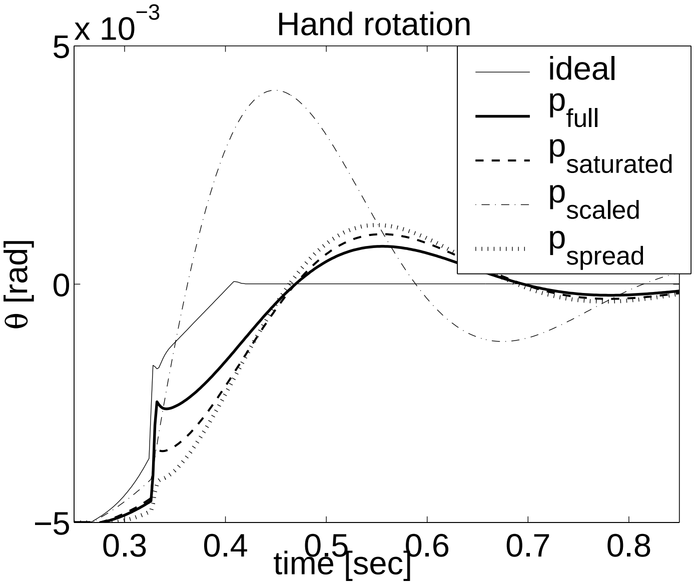
As the force capabilities of the haptic interface decrease, the haptic and visual performance of the proposed approach diminishes. For $e = -1$, the impulse-augmented penalty-based simulation reduces to a penalty-based simulation. The reduction in haptic performance can be seen by comparing Figures 4.15 and 4.19. In Figure 4.15, it is assumed that the device can fully apply the collision impulses to the user's hand. In Figure 4.19, it is assumed that the device can apply at most $\mathbf{F}_{limit} = (15\mathrm{N} \quad 15\mathrm{N} \quad 1\mathrm{Nm})^T$ and collision impulses are spread over more time steps when necessary. In this figure, "ideal", "IAPB LM", and "PB" represent interaction with a constraint-based, an impulse-augmented penalty-based, and a penalty-based environment, respectively, all connected to users through the four channel controller. "CB_{uc}" represents interaction with a constraint-based virtual world connected to users through the unilateral coupler.

The hand trajectories representing interactions with the impulse-augmented penalty-based and penalties-only virtual environments are closer to each other in Figure 4.19 than they are in Figure 4.15. The visual performance diminishes correspondingly, because the virtual tool penetrates the constraints deeper when spreading is required than when the device can fully apply the collision impulses. Nevertheless, constraint penetration is smaller in the proposed simulation than in the penalty-based one regardless of the device limitations. Both the haptic and the visual performance of the impulse-augmented penalty-based virtual world is better than that

[6] Equation (4.40) shows that the post-collision kinetic energy of the contact group is at most equal to its pre-collision kinetic energy for any $e \in [-1, 1]$.

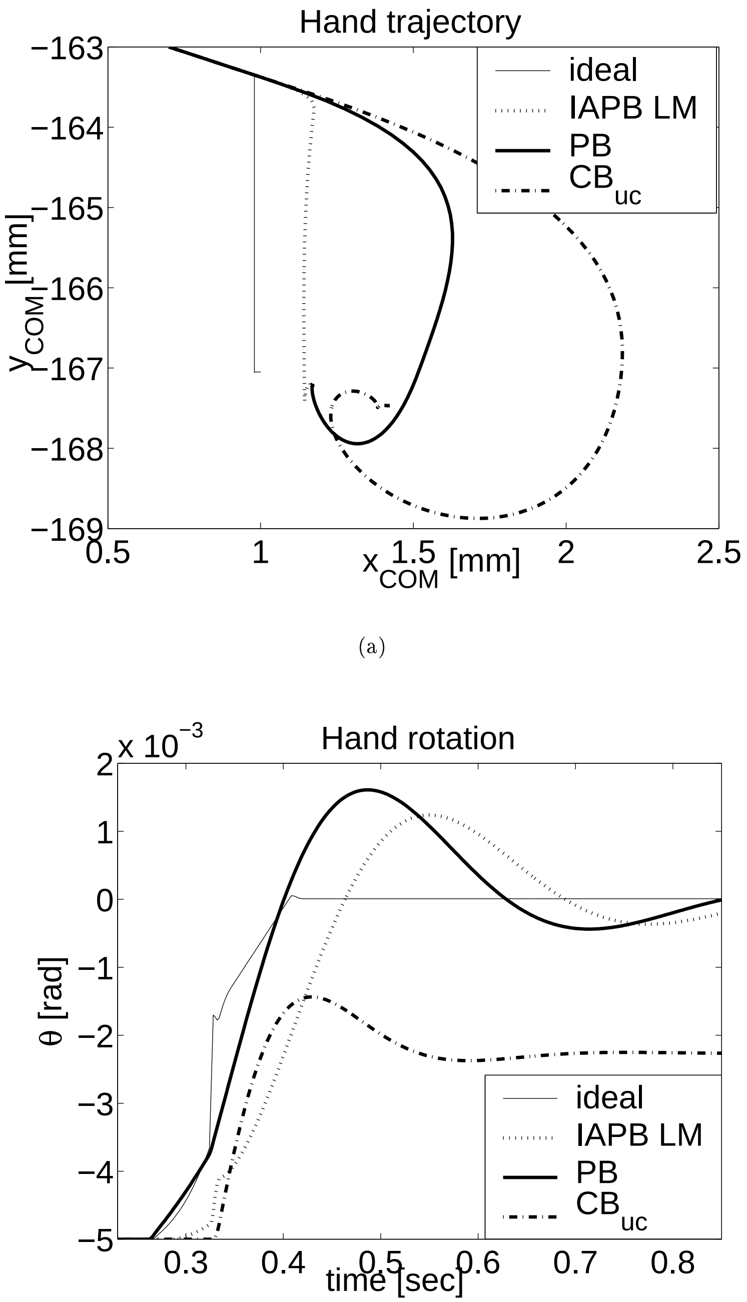


(a)



(b)

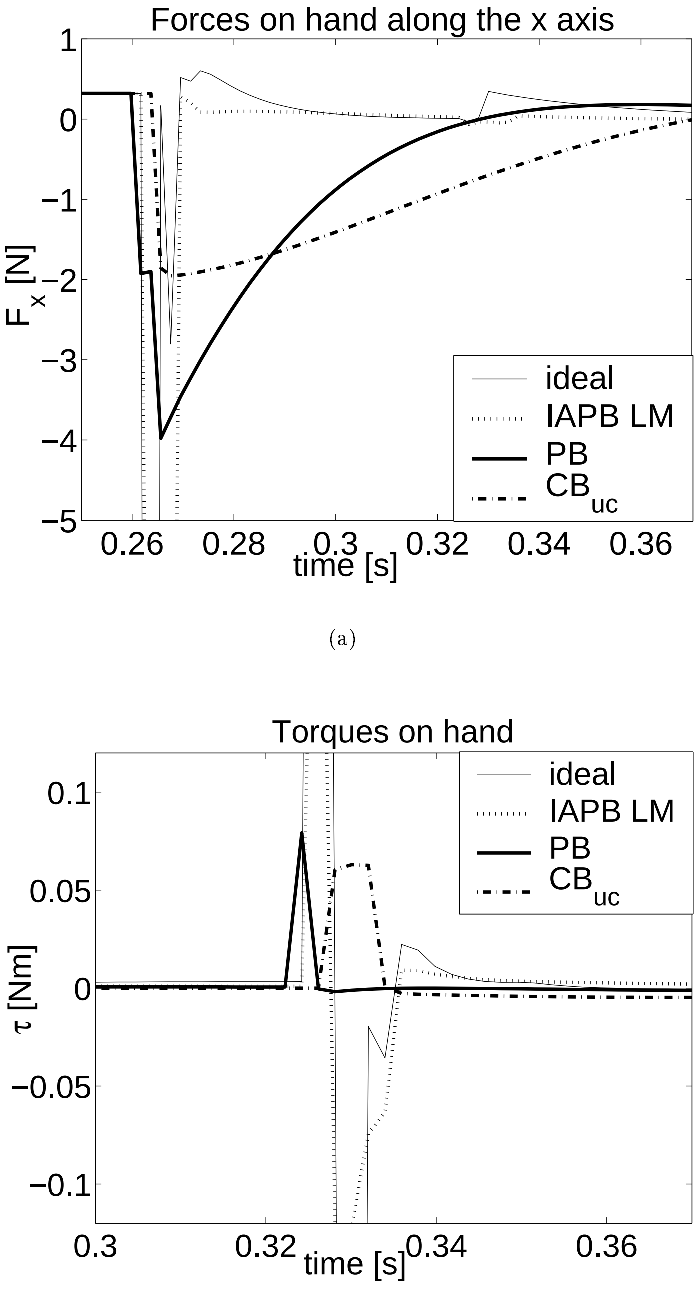
Figure 4.18: Simulated hand trajectories during user interaction with the impulse-augmented penalty-based virtual world when the device limitations are ignored and when they are taken into account ("ideal" - full constraint-based collision impulses are applied to users; "p_{full}" - full collision impulses are applied to users; "$p_{saturated}$" - collision impulses are saturated on the device; "p_{scaled}" - collision impulses are scaled in the virtual environment; "p_{spread}" - collision impulses are spread over several simulation steps when necessary).



(a)

(b)

Figure 4.19: Simulated hand trajectories obtained when constraint-based ("ideal"), impulse-augmented penalty-based ("IAPB LM"), and penalty-based ("PB") interactions are transmitted to users by a four channel controller, and when constraint-based interactions are transmitted to users by a unilateral coupler ("CB_{uc}"). Collision impulses computed using the proposed method are spread over several simulation steps when necessary.



(a)

(b)

Figure 4.20: Simulated forces and torques applied to users by the four channel controller when the virtual world is generated using the constraint-based ("ideal"), the impulse-augmented penalty-based ("IAPB LM") and the penalty-based ("PB") methods, and applied by the unilateral coupler when the virtual environment is generated using the constraint-based method ("CB_{uc}"). Collision impulses computed using the proposed technique are spread over several simulation steps when necessary.

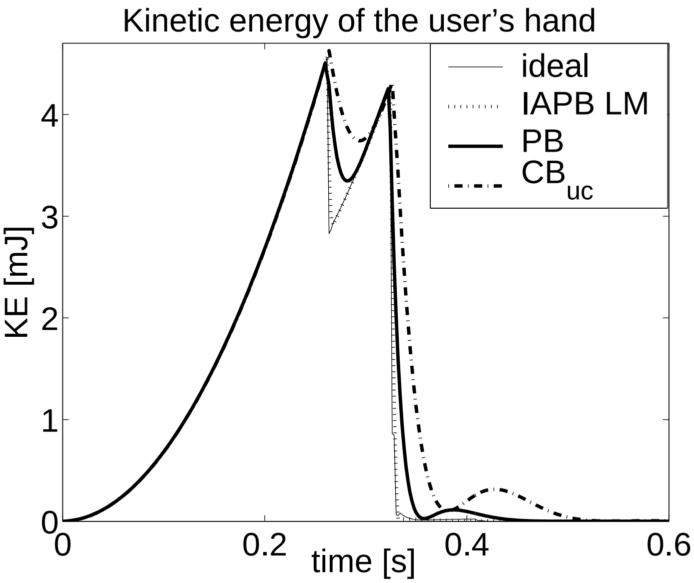


Figure 4.21: Kinetic energy of the users' hand when constraint-based ("ideal"), impulse-augmented penalty-based ("IAPB LM"), and penalty-based ("PB") interactions are applied to users by the four channel controller, and when constraint-based interactions are applied by the unilateral coupler ("CB$_{uc}$"). Collision impulses computed using the proposed method are spread over several simulation steps when necessary.

of the penalty-based world. In addition, the perceptual advantage obtained by applying abrupt forces to the user's hand upon contact and the increased stability of the interaction are maintained. Note that the wrenches felt by users are larger when users interact with the proposed simulation than when users interact with the penalty-based virtual environment (Figure 4.20). Furthermore, users lose more kinetic energy upon contact with the impulse-augmented penalty virtual constraints than upon contact with the penalty-based constraints (Figure 4.21). The increased contact passivity improves the coupled stability of the haptic interaction.

4.5.3 Experimental validation

In this section, the performance of the impulse-augmented penalty simulation approach is validated experimentally. The controlled interaction depicted in Figure 4.13 is implemented on the planar haptic simulation system described in Chapter 3 using the synchronous communication between the device and the virtual environment (see Figure 3.2(a)). Recall that the user's hand is represented through a constant wrench $\mathbf{F}_h = (0.32\text{N} \quad -0.4\text{N} \quad 0\text{Nm})^T$, collisions are considered perfectly plastic ($e = 0$), and the virtual walls have stiffness $K_{wall} = 4000\text{N/m}$ and damping $B_{wall} = 30\text{N/(m/s)}$. In the experiments, the virtual environment is generated using the penalty and the impulse-augmented penalty approaches[7]. Two implementations of the

[7] Implementations of haptic rigid body manipulations of constraint-based virtual environments running synchronously with the control loop are presently not available. This is because constraint-based simulations with guaranteed completion time have been developed only for perfectly plastic collisions between smooth convex objects [6]

impulse-augmented penalty approach are used: one considering that the device can fully apply the simulated impulses, and one accounting for device limitations.

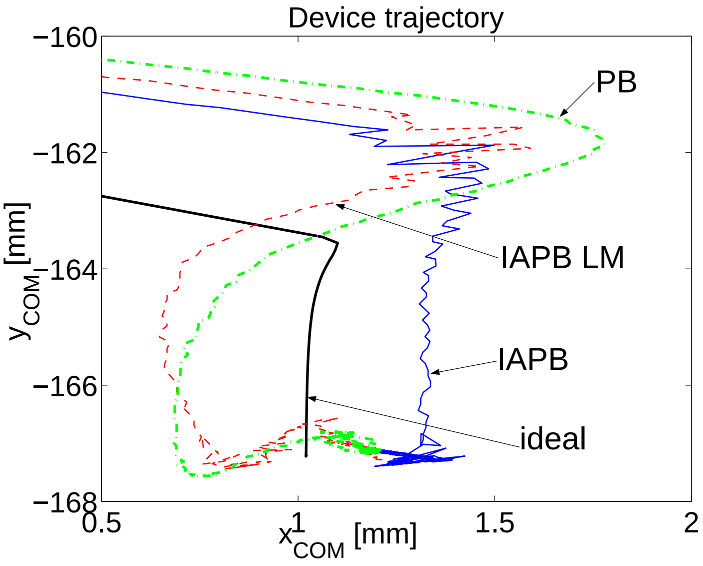
Figures 4.22, 4.23, and 4.24 monitor the device trajectories, the forces applied to users, and the kinetic energy of the device, respectively, when the peg is driven towards the corner, contacts the vertical virtual wall, slides down along it, contacts the horizontal virtual wall, and stops moving. The experimental results in these figures correspond to the cases when the virtual environment is generated using: (i) the impulse-augmented penalty approach and the device fully applies the simulated impulses to users, "IAPB"; (ii) the impulse-augmented penalty approach and the device applies at most $\mathbf{F}_{limit} = (15\text{N} \quad 15\text{N} \quad 1\text{Nm})^T$ to users, "IAPB LM"; and (iii) the penalty approach, "PB". The "ideal" trajectory is generated in Matlab^{TM} through simulating the interaction with a constraint-based virtual environment[8].

Figures 4.22 and 4.24 demonstrate the increased stability of the interaction in the impulse-augmented penalty virtual world compared to the penalty-based one. The peg loses more kinetic energy upon contact, settles into the corner faster, and bounces less when collision impulses are applied to the device. Hence, the impulsive forces applied upon constraint penetration improve the passivity of the virtual world and, thus, the stability of the haptic interaction. Moreover, they imrpove the realism of the interaction, because they generate a device trajectory that is closer to the trajectory imposed by a real rigid corner.

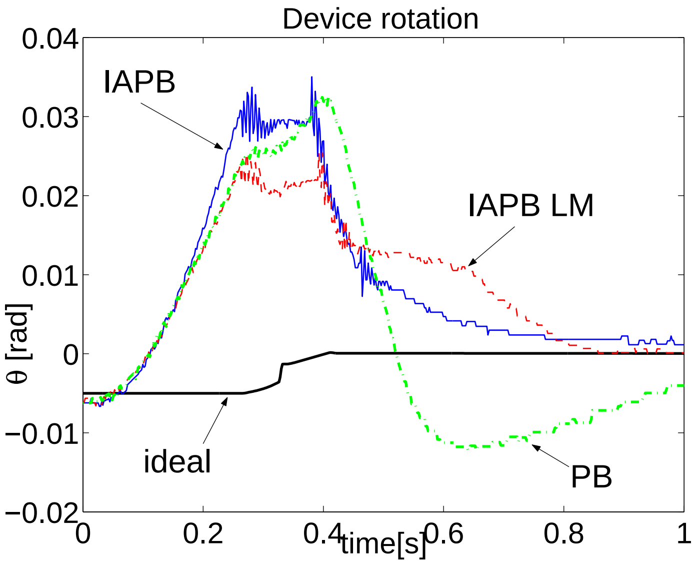
As predicted by simulations, the performance of the impulse-augmented penalty-based world decreases when collision impulses are applied over several steps in order to meet the device rendering capabilities. A device that can apply only limited forces and torques dissipates less energy upon impact than that predicted by the chosen coefficient of restitution. However, it dissipates more energy than during penalty-based interaction. Moreover, the perceptual advantage of large forces upon contact is maintained, because the impulsive forces rendered to users are much larger than the penalty-based contact forces despite the fact that they are limited by device capabilities (see Figure 4.23).

Note that increased virtual wall damping would also result in larger force transitions upon impact and less bouncing, i.e., more stable contact. However, the virtual damping is limited by the physical damping, the virtual wall stiffness, and the simulation step during 1 DOF interaction with a virtual wall [41]: the maximum allowable virtual damping decreases as the physical damping in the haptic interface decreases and as the wall stiffness and the simulation step increase. Hence, only limited improvements in the perceived rigidity of the virtual contacts could be achieved through increasing the virtual damping if the virtual wall stiffness is large (as needed for a convincingly rigid resting contacts) and the physical damping is small (as needed for imperceptible device dynamics). On the other hand, the impulsive forces provide a physically-based technique for enhancing the realism of the interaction that increases the stability of the interaction (without increasing the kinetic energy of the simulated environment) and is limited only by the device capabilities.

[8] Note that, due to friction in the interface, the "ideal" and the experimental trajectories have slightly different initial conditions. As a result, the virtual peg contacts the horizontal wall earlier in the simulation than in the experiments.

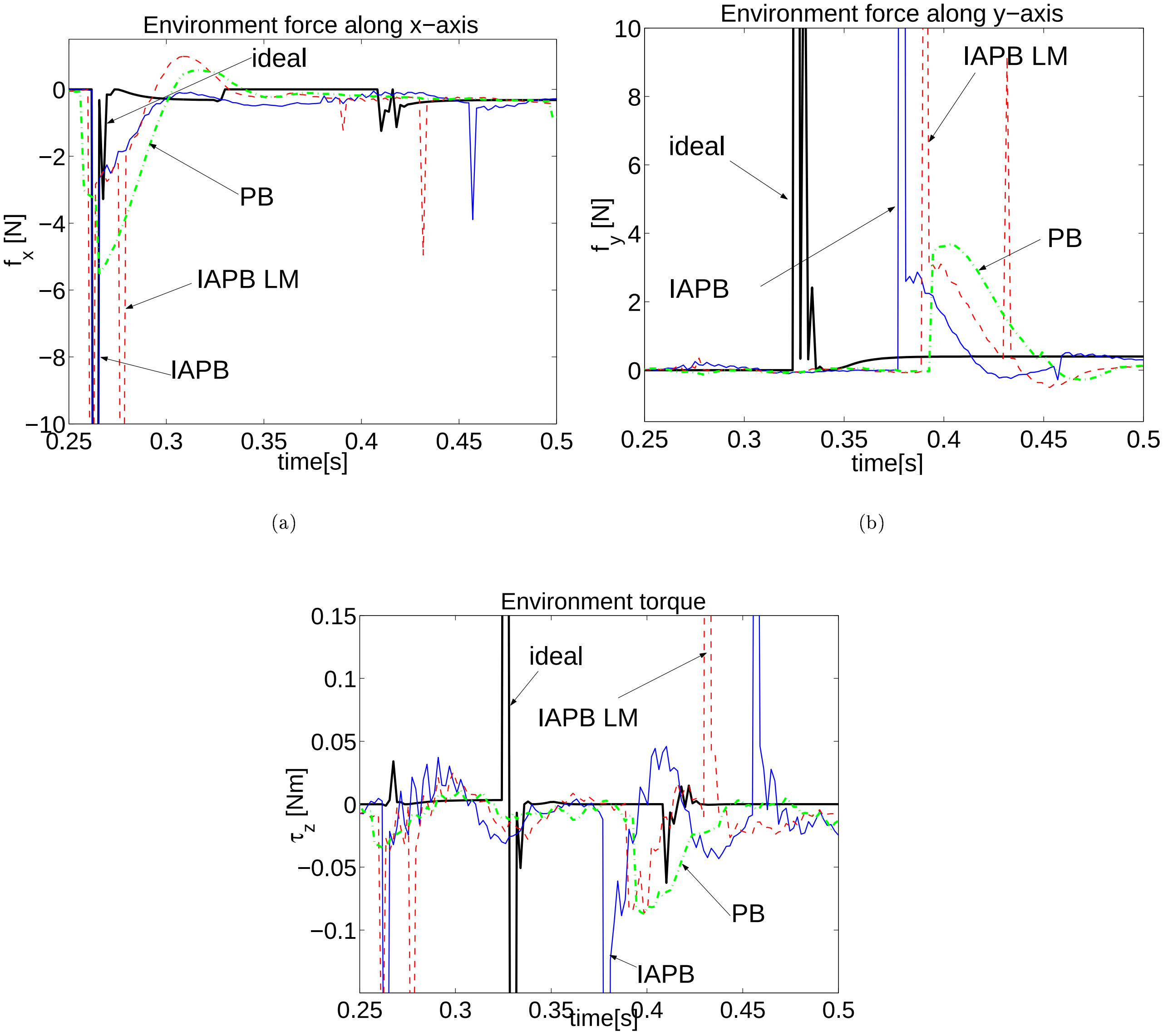


(a)



(b)

Figure 4.22: Device trajectories when the peg is pushed into the corner by a controller force. The virtual environment is generated using the impulse-augmented penalty-based and the penalty approaches ("IAPB" - impulse-augmented penalty-based world, full collision impulses applied to the device; "IAPB LM" - impulse-augmented penalty-based world, limited impulses applied to the device; "PB" - penalty world). The "ideal" trajectory is generated using a constraint-based simulation.



(a)

(b)

(c)

Figure 4.23: Environment wrenches applied to the device when the peg is pushed into the corner by a controlled force. The virtual environment is generated using the impulse-augmented penalty-based and the penalty approaches (“IAPB” - impulse-augmented penalty-based world, full collision impulses applied to the device; “IAPB LM” - impulse-augmented penalty-based world, limited impulses applied to the device; “PB” - penalty world). The “ideal” wrenches are generated using a constraint-based simulation.

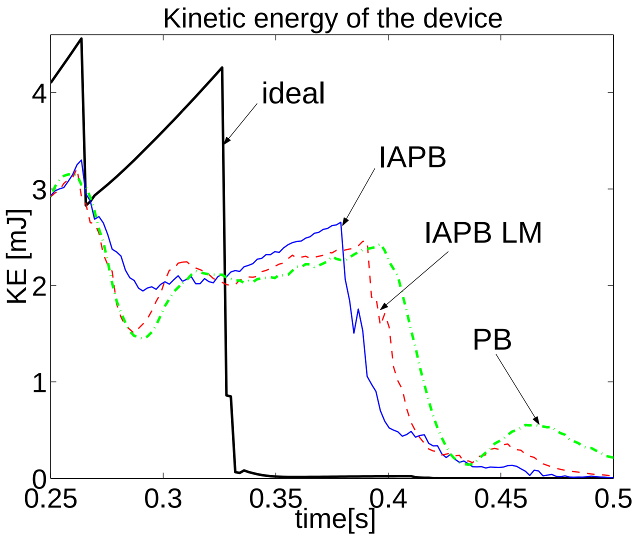


Figure 4.24: Kinetic energy of the device while it is pushed towards the corner by a controlled force. The virtual environment is generated using the impulse-augmented penalty-based and the penalty approaches (“IAPB” - impulse-augmented penalty-based world, full collision impulses applied to the device; “IAPB LM” - impulse-augmented penalty-based world, limited impulses applied to the device; “PB” - penalty world). The “ideal” wrenches are generated using a constraint-based simulation.

4.6 Discussion

The impulse-augmented penalty simulation approach introduced in this chapter has been designed to enable users to feel large forces upon contact with virtual objects and, in doing so, to increase the perceived rigidity of the virtual world [90, 125]. Compared to existing simulation paradigms, the approach allows haptic rendering of rigid body collisions, hence a more realistic representation of virtual contacts to users. Simulated and experimental interactions show that the selected friction model renders the stick-slip phenomenon similar to other models of dry friction used in haptics and that the selected model may potentially be used for changing the feel of sticking contacts. The simultaneous collision resolution never increases the kinetic energy of the virtual environment, thus maintaining the physical accuracy of the simulation and improving the stability of the virtual contacts. Moreover, it implicitly eliminates the overdetermined constraints, thereby increasing simulation efficiency. Inadequate force capabilities of the haptic interface degrade the performance of the approach. Performance degradation is limited through a novel technique that adapts the coefficient of restitution to the device capabilities without increasing the kinetic energy of the virtual environment. The new technique spreads the collision impulses over several steps of the simulation when the haptic interface cannot dissipate the kinetic energy prescribed by the desired coefficient of restitution during one simulation step.

Chapter 5

Manipulation of serial linkages

The previous chapter has proposed a simulation approach that computes impulsive forces upon contact and penalty-based and friction forces during contact. Virtual world dynamics have been formulated in configuration space, such that distinguishing between contacts of virtual objects and contacts of virtual linkages has not been necessary. This chapter discusses realistic kinesthetic operation of virtual linkages.

Realistic haptic manipulation of virtual linkages depends on: (i) the physical accuracy of the linkage simulation; (ii) the user's perception of the linkage contacts; and (iii) the user's perception of the linkage dynamics and of the topological constraints. Since linkage dynamics are challenging to simulate at the haptic rates, prior work focused on developing both application-specific [114] and general-purpose [126–128] techniques that address the first two factors. In particular, efficient dynamics are the main concern in [114, 126, 128] and accurate collision/contact models are developed in [127]. In all cases, only penalty-like forces are applied to users. Moreover, operation of linkages only from links with redundant degrees of freedom is allowed.

In this chapter, simulation and control techniques are introduced that enable realistic operation of linkages from any user-selected link and through singularities. The chapter starts by discussing the control architecture employed to render convincing force interactions between a virtual linkage and a rigid multibody virtual environment. A coordinate-invariant representation of contact forces/impulses at the user's hand is then introduced. The control methods that allow users to convincingly perceive the inertia and the topology of the virtual linkage follow. Combined with the impulse-augmented simulation approach presented in the previous chapter, these techniques permit realistic and unrestricted haptic manipulation of serial linkages. Throughout this chapter, it is assumed that users manipulate a virtual linkage, i.e., virtual tool and virtual linkage are used interchangeably. The schematic of haptic manipulation of virtual linkages within an impulse-augmented virtual world is shown in Figure 5.1. For simplicity, the position coordination signals used by the haptic controller are not represented. In this figure, $\mathbf{F}_{constr}$, $\mathbf{F}_{inertia}$, $\mathbf{F}_h$, and $\mathbf{F}_{env}$ are the wrenches due to the topological constraints, the linkage inertia, the user, and the contact forces/impulses that act on the virtual tool, respectively, all represented at the user's hand. $\mathbf{f}_1$, and $\mathbf{f}_2$ are the contact forces/impulses that act between the virtual linkage and other virtual objects, $\mathbf{n}_{tc}$ represents the configuration space directions of constraint imposed on the user's motion by the linkage topology, $\mathbf{\Lambda}_h^{-1}$ is the inverse of the operational space inertia of the linkage [80] at the user's hand, and $\mathbf{T}_{env}$ is the configuration space torque due to the contacts between the virtual tool and the virtual environment.

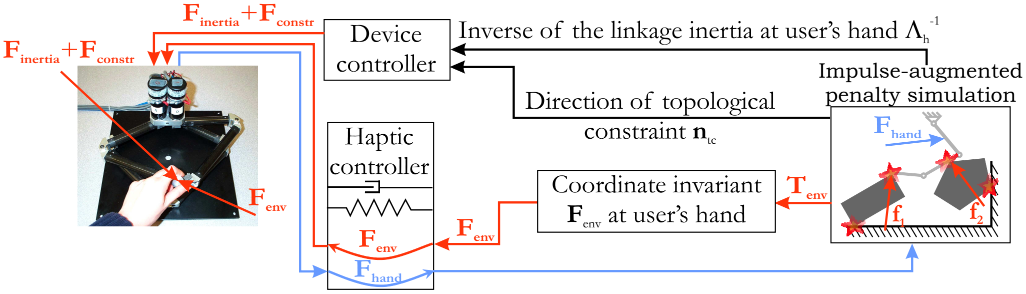

$\mathbf{F}_{inertia}+\mathbf{F}_{constr}$
$\mathbf{F}_{inertia}+\mathbf{F}_{constr}$
Device controller
Inverse of the linkage inertia at user's hand Λ_h^{-1}
Impulse-augmented penalty simulation
Direction of topological constraint $\mathbf{n}_{tc}$
Haptic controller
$\mathbf{F}_{env}$
Coordinate invariant $\mathbf{F}_{env}$ at user's hand
$\mathbf{T}_{env}$
$\mathbf{F}_{hand}$
$\mathbf{f}_1$
$\mathbf{f}_2$
$\mathbf{F}_{env}$
$\mathbf{F}_{env}$
$\mathbf{F}_{hand}$

Figure 5.1: Realistic haptic manipulation of virtual linkages in an impulse-augmented penalty virtual world.

5.1 Control architecture for realistic linkage manipulation

Besides the linkage simulation, both the device controller and the haptic controller contribute to realistic force interaction within the virtual environment during linkage manipulation. The device controller [136] uses impedance control [68] to apply the inertia of the linkage to the user's hand. In particular, it changes the impedance of the haptic device to match the impedance of the virtual tool. The haptic controller [136] uses a four channel architecture to apply the contact forces/impulses that act on the virtual tool to the user's hand (see discussion in Section 4.1).

A 1 DOF mechanical equivalent of this control architecture (excluding the communication delay) is depicted in Figure 5.2. In this figure, Z_d and Z_{vt} are the impedance of the haptic device and of the virtual tool, respectively, f_h, f_{env}, and f_{pc} are the hand, the environment, and the coordination force[1], respectively, and S scales the two force channels of the haptic controller (from the device to the virtual environment and vice-versa). If the device impedance equals the virtual tool impedance, $Z_d = Z_{vt}$, and forces are not scaled, $S = 1$, the haptic controller is transparent. Users perceive the inertia and the topology of the linkage through the virtual tool impedance Z_{vt}. They perceive the motion constraints imposed by other virtual objects through the environment force f_{env}. Hence, the transparency of the haptic controller and the accurate reflection of the virtual tool impedance through device control are important for realistic haptic manipulation of linkages.

Although not apparent in the 1 DOF mechanical equivalent of the control architecture, the representation at the user's hand of the contact forces/impulses acting on the linkage also affects the realism of the interaction. This representation is discussed in the following section. Haptic rendering of the linkage inertia and of the motion constraints imposed on users by the linkage topology are presented in later sections.

[1] The coordination force is applied by the position coordination channels of the haptic controller. These channels are shown in Figure 5.2 as a spring-damper connection between the device and the virtual tool.

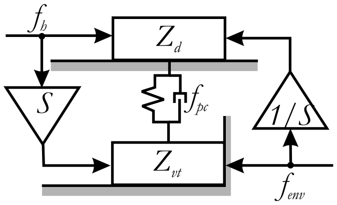


Figure 5.2: 1 DOF mechanical equivalent of the control architecture.

5.2 Linkage contacts at user's hand

The rigid multibody simulation approach developed in Chapter 4 evolves the virtual environment dynamics in configuration space. In this space, virtual objects are indistinguishable from virtual linkages. However, users interact with the virtual environment in Cartesian (operational [80]) space through a haptic controller that applies the simulated forces/impulses to their hand (see Figure 5.1). Hence, the contact forces/impulses acting on the virtual tool must be suitably represented at the user's hand during virtual linkage manipulation[2]. A coordinate invariant representation of these contact forces/impulses at the user-selected link is proposed in this section.

The user-applied wrench $\mathbf{F}_h$ is mapped to the hand configuration torque $\mathbf{T}_h$ by the transpose of the Jacobian of the virtual linkage computed at the user's hand, $\mathbf{J}_h^T$, herein called the "hand Jacobian":

$$\mathbf{T}_h = \mathbf{J}_h^T \mathbf{F}_h. \tag{5.1}$$

Hence, $\mathbf{J}_h^T$ maps the space F of wrenches at the user's hand to the space T of configuration torques. Conversely, the space of configuration torques must be mapped to the space of wrenches at the user's hand in order to allow the haptic controller to apply the contact forces/impulses that act on the virtual linkage to the user.

When $\mathbf{J}_h^T$ is invertible, the required mapping is provided by the transpose of the inverse of the virtual tool Jacobian at the user's hand:

$$\mathbf{F}_{env} = (\mathbf{J}_h)^{-T} \mathbf{T}_{env}. \tag{5.2}$$

In Equation (5.2), $\mathbf{T}_{env}$ is the configuration space representation of the contact forces/impulses acting on the virtual tool (i.e., the environment configuration torque), and $\mathbf{F}_{env}$ is its wrench representation at the user's hand (i.e., the environment wrench). According to the derivations in Chapter 4:

$$\mathbf{T}_{env} = \sum_{i=1}^{c_{vt}} \mathbf{J}_i^T \mathbf{f}_i, \tag{5.3}$$

during resting contact, and:

$$\mathbf{T}_{env} = \sum_{i=1}^{c_{vt}} \mathbf{J}_i^T \mathbf{p}_i = \mathcal{J}_{c_{vt}}^T \mathbf{p}, \tag{5.4}$$

[2] Cartesian (operational) space coincides with configuration space during manipulation of virtual objects and the mapping of contact forces/impulses to configuration space through the contact Jacobians $\mathbf{J}_i$ suffices.

during colliding contact. In Equations (5.3) and (5.4), c_{vt} is the number of contacts between the virtual tool and the virtual environment (assuming a suitable numbering of the contacts of the contact group to which the virtual tool belongs), and $\mathcal{J}_{c_{vt}}^T = \begin{bmatrix}\mathbf{J}_1^T\mathbf{n}_1 & \dots & \mathbf{J}_i^T\mathbf{n}_i & \dots & \mathbf{J}_{c_{vt}}^T\mathbf{n}_{c_{vt}}\end{bmatrix}^T$.

During unrestricted haptic manipulation of arbitrary linkages, $\mathbf{J}_h^T$ may not be invertible. This is because: (i) the linkage may have any number of degrees of freedom (in particular, more or less than 6DOF); (ii) the user may choose to operate it from any link (from the one proximal to the fixed base to the one distal to the base); and (iii) the user may move the linkage through singularities. In all these situations, the configuration torques must be mapped to hand wrenches such that the results are physically significant, i.e., gauge invariant under rigid body transformations [46].

The mapping of hand wrenches to simulated configuration torques given in Equation (5.1) and the mapping of simulated configuration velocities $\dot{\mathbf{q}}$ to simulated hand twists $\dot{\mathbf{x}}_{h_{vt}}$:

$$\dot{\mathbf{x}}_{h_{vt}} = \mathbf{J}_h\dot{\mathbf{q}} \tag{5.5}$$

form a dual system [46] (see Appendix B for definition). In Equation (5.5), $\dot{\mathbf{x}}_{h_{vt}} = \begin{bmatrix}\mathbf{v}_h^T & \boldsymbol{\omega}^T\end{bmatrix}^T$, where $\mathbf{v}_h \in \mathcal{R}^3$ and $\boldsymbol{\omega} \in \mathcal{R}^3$ are the linear velocity of the user-selected operational point on the virtual tool and the angular velocity of the virtual link held by the user, respectively. Therefore, as discussed in detail in [46] and briefly summarized in Appendix B, a mapping from T to F is provided by the transpose of the weighted generalized inverse of $\mathbf{J}_h$ [46]:

$$\mathbf{J}_h^{\#} = \mathbf{M}_Q^{-1}\mathbf{C}^T\left(\mathbf{C}\mathbf{M}_Q^{-1}\mathbf{C}^T\right)^{-1}\left(\mathbf{F}^T\mathbf{M}_V\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{M}_V. \tag{5.6}$$

In Equation (5.6), $\mathbf{J}_h^{\#}$ is a weighted generalized inverse of the hand Jacobian, $\mathbf{J}_h = \mathbf{FC}$ is a full-rank decomposition (factorization) [20] of the hand Jacobian $\mathbf{J}_h$[3], $\mathbf{M}_Q$ is the metric on the space Q of configuration velocities, and $\mathbf{M}_V$ is the metric on the space V of hand twists. Coordinate invariance of the mapping $\mathbf{J}_h^{\#}$ is ensured through a suitable choice of metrics on the spaces Q and V [46]. In [46] it is shown that kinetic energy metrics are gauge invariant. In this work, the metric on Q is provided by the configuration space mass matrix of the virtual tool $\mathbf{D}_{vt}$. The metric on V is provided by the mass matrix of the link held by the user $\mathbf{M}_{hl}$, computed in the link's coordinate frame. Hence, the coordinate invariant weighted generalized inverse of the Jacobian of the virtual tool computed at the user's hand is given by:

$$\mathbf{J}_h^{\#} = \mathbf{D}_{vt}^{-1}\mathbf{C}^T\left(\mathbf{C}\mathbf{D}_{vt}^{-1}\mathbf{C}^T\right)^{-1}\left(\mathbf{F}^T\mathbf{M}_{hl}\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{M}_{hl}. \tag{5.7}$$

The properties of $\mathbf{J}_h^{\#}$ are presented in Appendix B. The algorithm proposed for computing a full rank factorization of $\mathbf{J}_h$ is detailed in the following. The algorithm uses the Singular Value Decomposition (SVD) of $\mathbf{J}_h$:

$$\mathbf{J}_{h_{6\times d}} = \mathbf{U}_{6\times 6}\boldsymbol{\Sigma}_{6\times d}\mathbf{V}_{d\times d}^T. \tag{5.8}$$

[3] A full-rank decomposition [46] $\mathbf{J}_h = \mathbf{FC}$ satisfies $\mathit{rank}\,(\mathbf{J}_h) = \mathit{row\text{-}rank}\,(\mathbf{C}) = \mathit{column\text{-}rank}\,(\mathbf{F})$.

For a hand Jacobian with $rank\left(\mathbf{J}_h\right) = r$, only the first r diagonal elements of the matrix $\mathbf{\Sigma}$ are non-zero and:

$$\mathbf{\Sigma}_{6\times d} = \begin{bmatrix} \mathbf{\Sigma}'_{r\times r} & \mathbf{0}_{r\times(d-r)} \\ \mathbf{0}_{(6-r)\times r} & \mathbf{0}_{(6-r)\times(d-r)} \end{bmatrix} = \begin{bmatrix} \mathbf{\Sigma}'_{r\times r} \\ \mathbf{0}_{(6-r)\times r} \end{bmatrix} \begin{bmatrix} \mathbf{I}_{r\times r} & \mathbf{0}_{r\times(d-r)} \end{bmatrix} = \mathbf{\Sigma}_{1_{6\times r}} \mathbf{\Sigma}_{2_{r\times d}}, \tag{5.9}$$

where $\mathbf{I}_{r\times r}$ is the r-dimensional unit matrix and $\mathbf{0}_{(6-r)\times r}$ denotes a matrix of zeros with the given dimensions. A full rank factorization of $\mathbf{J}_h$ is then given by:

$$\begin{aligned} \mathbf{F} &= \mathbf{U}\mathbf{\Sigma}_1 \\ \mathbf{C} &= \mathbf{\Sigma}_2\mathbf{V}^T \end{aligned}, \tag{5.10}$$

and the coordinate invariant inverse of the hand Jacobian for the chosen set of metrics is:

$$\mathbf{J}_h^{\#} = \mathbf{D}_{vt}^{-1}\mathbf{V}\mathbf{\Sigma}_2^T\left(\mathbf{\Sigma}_2\mathbf{V}^T\mathbf{D}_{vt}^{-1}\mathbf{V}\mathbf{\Sigma}_2^T\right)^{-1}\left(\mathbf{\Sigma}_1^T\mathbf{U}^T\mathbf{M}_{hl}\mathbf{U}\mathbf{\Sigma}_1\right)^{-1}\mathbf{\Sigma}_1^T\mathbf{U}^T\mathbf{M}_{hl}. \tag{5.11}$$

Hence, the environment configuration torque is mapped to the environment wrench $\mathbf{F}_{env}$ according to:

$$\mathbf{F}_{env} = \mathbf{J}_h^{\#^T}\mathbf{T}_{env}. \tag{5.12}$$

5.3 Haptic rendering of linkage inertia

In Cartesian (operational [80]) space, both the inertia and the topological constraints are embedded in the operational space inertia of the virtual linkage computed at the user's hand [80], $\mathbf{\Lambda}_h$:

$$\mathbf{\Lambda}_h = \left(\mathbf{J}_h\mathbf{D}_{vt}^{-1}\mathbf{J}_h^T\right)^{-1}. \tag{5.13}$$

Therefore, $\mathbf{\Lambda}_h$ is incorporated in the desired impedance of the virtual tool Z_{vt}. Specifically, the device controller changes the dynamics of the device to match the desired dynamics of the virtual tool [136]:

$$\mathbf{\Lambda}_h\ddot{\mathbf{x}}_h + \mathbf{b}_d\dot{\mathbf{x}}_h + \mathbf{k}_d\mathbf{x}_h = \mathbf{F}_h + \mathbf{F}_{env} + \mathbf{F}_{pc}. \tag{5.14}$$

In Equation (5.14), $\mathbf{\Lambda}_h \in \mathcal{R}^{6\times 6}$, $\mathbf{b}_d \in \mathcal{R}^{6\times 6}$, and $\mathbf{k}_d \in \mathcal{R}^{6\times 6}$ are the desired (i.e., simulated) inertia, damping, and stiffness of the virtual tool, respectively, $\mathbf{F}_{pc} \in \mathcal{R}^6$ is the position coordination wrench between the device and the virtual tool (due to the generalized spring-damper shown acting between them in Figure 5.2), and $\ddot{\mathbf{x}}_h \in \mathcal{R}^6$, $\dot{\mathbf{x}}_h \in \mathcal{R}^6$, and $\mathbf{x}_h \in \mathcal{R}^6$ are the desired body acceleration, velocity, and position of the device at the user-selected link, respectively.

Equation (5.14) is equivalent to:

$$\ddot{\mathbf{x}}_h = \mathbf{\Lambda}_h^{-1}\left(\mathbf{F}_h + \mathbf{F}_{env} + \mathbf{F}_{pc} - \mathbf{b}_d\dot{\mathbf{x}}_h - \mathbf{k}_d\mathbf{x}_h\right). \tag{5.15}$$

This form is advantageous because the inverse of the operational space inertia of the virtual tool at the user's hand can be computed regardless of the rank of the Jacobian:

$$\mathbf{\Lambda}_h^{-1} = \mathbf{J}_h \mathbf{D}_{vt}^{-1} \mathbf{J}_h^T. \tag{5.16}$$

In contrast, $\mathbf{\Lambda}_h$ can be computed only when the Jacobian is full row-rank. When $\mathbf{J}_h$ is not full row-rank, $\mathbf{\Lambda}_h^{-1}$ becomes infinite along certain directions of the operational space and $\mathbf{\Lambda}_h^{-1}$ drops rank.

Rank deficiency of $\mathbf{\Lambda}_h^{-1}$ indicates that the topology of the virtual tool restricts the simulated instantaneous motion of the user. This may happen when the user holds a link with fewer than 6 DOFs during spatial interaction and fewer than 3 DOFs during planar interaction or when they operate the virtual linkage through a singularity. To illustrate how the virtual motion constraints imposed on users by the linkage topology are represented in the rank deficiency of $\mathbf{\Lambda}_h^{-1}$, consider the example manipulations of the planar linkage depicted in Figure 5.3. In the first example, the user holds the linkage from the centre of mass of the middle link in the position shown in Figure 5.3(a). The held link has insufficient degrees of freedom to allow arbitrary position and orientation at the user-selected operational point. Hence, the linkage topology instantaneously constrains the user's motion according to:

$$\dot{\mathbf{x}}_h = \mathbf{J}_h \dot{\mathbf{q}}, \tag{5.17}$$

where the hand Jacobian is:

$$\mathbf{J}_h = \begin{bmatrix} -l_1 \sin(q_1) - l_{c_2} \sin(q_1 + q_2) & -l_{c_2} \sin(q_1 + q_2) & 0 \\ l_1 \cos(q_1) + l_{c_2} \cos(q_1 + q_2) & l_{c_2} \cos(q_1 + q_2) & 0 \\ 1 & 1 & 0 \end{bmatrix}. \tag{5.18}$$

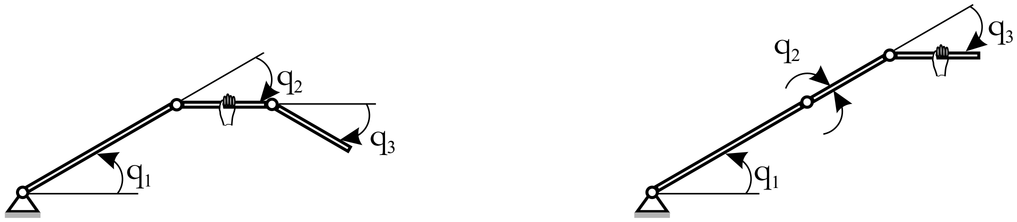
Equation (5.18) shows that, due to the topological constraint, $rank(\mathbf{J}_h) = 2$. From Equation (5.16), it follows that $rank\left(\mathbf{\Lambda}_h^{-1}\right) = 2$. In the second example, the user holds the linkage from the centre of mass of the distal link in the position shown in Figure 5.3(b). In the example position, the Jacobian:

$$\mathbf{J}_h = \begin{bmatrix} -l_1 \sin(q_1) - l_2 \sin(q_1) - l_{c_3} \sin(q_1 + q_3) & -l_2 \sin(q_1) - l_{c_3} \sin(q_1 + q_3) & -l_{c_3} \sin(q_1 + q_3) \\ l_1 \cos(q_1) + l_2 \cos(q_1) + l_{c_3} \cos(q_1 + q_3) & l_2 \cos(q_1) + l_{c_3} \cos(q_1 + q_3) & l_{c_3} \cos(q_1 + q_3) \\ 1 & 1 & 1 \end{bmatrix}, \tag{5.19}$$

is singular, $rank(\mathbf{J}_h) = 2$. Consequently, $rank\left(\mathbf{\Lambda}_h^{-1}\right) = 2$. In other words, $\mathbf{\Lambda}_h^{-1}$ drops rank when the linkage restricts the user's motion.

Since $\mathbf{\Lambda}_h^{-1}$ can be computed regardless of the rank of the hand Jacobian, Equation (5.15) is used for the impedance control of the device. This equation computes the desired body acceleration at the user's hand. The dynamics of the haptic interface are given by:

$$\mathbf{M}_d \ddot{\mathbf{x}}_h + \mathbf{C}_d \dot{\mathbf{x}}_h = \mathbf{F}_h + \mathbf{u}, \tag{5.20}$$



(a) Manipulation from a link proximal to the base.

(b) Manipulation through a singularity.

Figure 5.3: Example manipulations restricted by the topology of the virtual linkage.

where $\mathbf{M}_d$ and $\mathbf{C}_d$ are the inertia and Christoffel matrices of the device and $\mathbf{u}$ is the control signal [136]. By combining Equations (5.15) and (5.20), the impedance control law for the interface is obtained as:

$$\mathbf{u} = \left(\mathbf{M}_d\boldsymbol{\Lambda}_h^{-1} - \mathbf{I}\right)\mathbf{F}_h + \mathbf{M}_d\boldsymbol{\Lambda}_h^{-1}\left(\mathbf{F}_{env} + \mathbf{F}_{pc}\right) + \left(\mathbf{C}_d - \mathbf{M}_d\boldsymbol{\Lambda}_h^{-1}\mathbf{b}_d\right)\dot{\mathbf{x}}_h - \mathbf{M}_d\boldsymbol{\Lambda}_h^{-1}\mathbf{k}_d\mathbf{x}_h. \tag{5.21}$$

Equation (5.21) applies the inertia and the topology of the virtual tool to users at the acceleration level through $\boldsymbol{\Lambda}_h^{-1}$. Due to numerical drift and limited device stiffness, Equation (5.21) is not sufficient for realistic haptic rendering of the topological constraints. A technique that enables users to convincingly perceive these constraints is presented in the following section.

5.4 Haptic rendering of linkage topology

An analysis of the virtual tool dynamics suggests how the control in Equation (5.21) can be augmented to enable users to accurately perceive linkage topology. Consider that, at the user-selected operational point, the environment wrench is $\mathbf{F}_{env}$ and the inverse of the virtual tool operational space inertia is $\boldsymbol{\Lambda}_h^{-1}$. This inverse maps the environment wrench to the body acceleration at the user's hand:

$$\ddot{\mathbf{x}}_h = \boldsymbol{\Lambda}_h^{-1}\mathbf{F}_{env}. \tag{5.22}$$

In Equation (5.22), $\ddot{\mathbf{x}}_h$ is an element of the space of body accelerations at the user's hand, A; i.e., $\ddot{\mathbf{x}}_h = \left(\dot{\mathbf{v}}_h^T \quad \dot{\boldsymbol{\omega}}^T\right)^T$, where $\dot{\mathbf{v}}_h \in \mathcal{R}^3$ and $\dot{\boldsymbol{\omega}} \in \mathcal{R}^3$ are the linear and the angular acceleration of the user's hand, respectively. When $\boldsymbol{\Lambda}_h^{-1}$ is full rank, the environment wrench produces body acceleration of the user's hand along all directions of A. In other words, the virtual tool behaves as an inertia along all directions of A. On the control side, Equation (5.21) applies this inertia to users by controlling their acceleration to the value given in Equation (5.22).

When $\boldsymbol{\Lambda}_h^{-1}$ is rank-deficient, $\ddot{\mathbf{x}}_h$ lies in a subspace of A. Then, environment wrenches exist that lie in the null space of $\boldsymbol{\Lambda}_h^{-1}$, $\mathcal{N}\left(\boldsymbol{\Lambda}_h^{-1}\right)$ (hereafter called the space of wrenches of topological constraint). These wrenches are directly opposed by the structural stiffness of the virtual linkage and have no effect on the body acceleration at the user's hand. Hence, at the user-selected operational point, the linkage behaves as an infinite stiffness along the directions of topological constraint and behaves as an inertia along the directions of topological freedom of motion[4]. On the control side, the impedance control in Equation (5.21) is unsuitable for rendering the infinite structural stiffness of the virtual tool along the directions of topological constraint.

Realistic perception of the topological constraints is enabled through augmenting the control provided by Equation (5.21) with penalties applied along the singular directions of $\mathbf{\Lambda}_h^{-1}$, as shown in Figure 5.4. In this figure, vt is the user's position and orientation on the virtual tool (i.e., in the simulation), while h is their real position and orientation. For simplicity, only a one dimensional null space of $\mathbf{\Lambda}_h^{-1}$ is depicted.

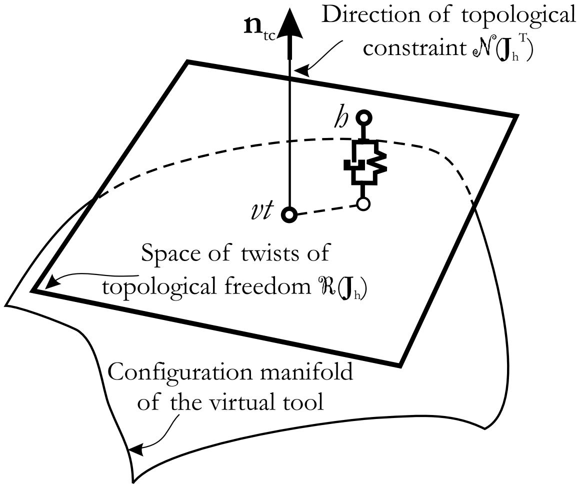

$\mathbf{n}_{tc}$
Direction of topological constraint $\mathcal{N}(\mathbf{J}_h^T)$
h
vt
Space of twists of topological freedom $\mathcal{R}(\mathbf{J}_h)$
Configuration manifold of the virtual tool

Figure 5.4: Penalty wrench constraining users to the configuration manifold of the virtual linkage that they manipulate.

The directions of topological constraint $\mathbf{n}_{tc}$ are provided by the null space of $\mathbf{\Lambda}_h^{-1}$. Note that the maximum number of such directions is five during spatial rigid body (6 DOFs) manipulation and two during planar rigid body (3 DOFs) manipulation. The constraint position and velocity are provided by the body position $\mathbf{x}_{h_{vt}}$ and velocity $\dot{\mathbf{x}}_{h_{vt}}$ of the user-selected operational point on the virtual tool. Then, given the stiffness k_{tc} and damping b_{tc} of the topological constraints, the impedance control law in Equation (5.21) is modified according to:

$$\begin{aligned}\mathbf{u} &= \mathbf{M}_d\left(\ddot{\mathbf{x}}_{h_{vt}} + \sum_{i=1}^{n_{tc}}\left(k_{tc}\mathbf{n}_{tc,i}^T\left(\mathbf{x}_{h_{vt}} - \mathbf{x}_h\right) + b_{tc}\mathbf{n}_{tc,i}^T\left(\dot{\mathbf{x}}_{h_{vt}} - \dot{\mathbf{x}}_h\right)\right)\mathbf{n}_{tc,i}\right) + \mathbf{C}_d\dot{\mathbf{x}}_h - \mathbf{F}_h = \\ &= \left(\mathbf{M}_d\mathbf{\Lambda}_h^{-1} - \mathbf{I}\right)\mathbf{F}_h + \mathbf{M}_d\mathbf{\Lambda}_h^{-1}\left(\mathbf{F}_{env} + \mathbf{F}_{pc}\right) + \left(\mathbf{C}_d - \mathbf{M}_d\mathbf{\Lambda}_h^{-1}\mathbf{b}_d\right)\dot{\mathbf{x}}_h - \mathbf{M}_d\mathbf{\Lambda}_h^{-1}\mathbf{k}_d\mathbf{x}_h \\ &\quad + \mathbf{M}_d\sum_{i=1}^{n_{tc}}\left(k_{tc}\mathbf{n}_{tc,i}^T\left(\mathbf{x}_{h_{vt}} - \mathbf{x}_h\right) + b_{tc}\mathbf{n}_{tc,i}^T\left(\dot{\mathbf{x}}_{h_{vt}} - \dot{\mathbf{x}}_h\right)\right)\mathbf{n}_{tc,i},\end{aligned} \tag{5.23}$$

where n_{tc} is the number of topological constraints, i.e., $n_{tc} = 6 - rank(\mathbf{\Lambda}^{-1})$.

Note that no additional numerical effort (beyond computing the coordinate invariant representation of the contact forces/impulses at the user's hand) is necessary to derive the directions of topological constraint $\mathbf{n}_{c,i}$.

[4] The directions of topological freedom of motion form a basis on $\mathcal{R}(\mathbf{J}_h)$, the range space of $\mathbf{J}_h$.

This is because the singular directions of $\mathbf{\Lambda}_h^{-1}$ are the same as the singular directions of $\mathbf{J}_h^T$. The proof is given in Appendix C. Physically, the proof can be understood by considering the actions of $\mathbf{J}_h^T$ and of $\mathbf{\Lambda}_h^{-1}$. In particular, the user-applied torques balanced by the structural stiffness of the linkage are filtered out from the virtual tool configuration space dynamics by $\mathbf{J}_h^T$. $\mathbf{\Lambda}_h^{-1}$ filters out the same torques from operational space. Hence, the two mappings have equal null spaces and the directions of topological constraint are a byproduct of the full rank factorization of $\mathbf{J}_h$.

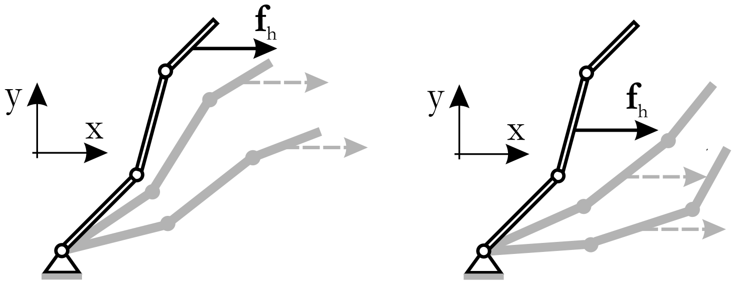
The ability of the control law in Equation (5.23) to enforce linkage topology is validated through simulations in the following section and through experiments in Section 5.4.2.

5.4.1 Simulations

In the simulations, users manipulate a three links planar linkage with rotational joints. Its parameters are given in Table 5.1. Initially, the virtual linkage is at rest in the position $\mathbf{q} = \left(\frac{\pi}{4}\text{rad} \quad \frac{\pi}{6}\text{rad} \quad -\frac{\pi}{6}\text{rad}\right)^T$ (see Figure 5.5). The user applies a constant force $f_h = 1\text{N}$ along the x direction. The device is also at rest, but its position differs from that of the user-selected operational point by 5 mm along the x direction and by 3 mm along the y direction.

Table 5.1: Parameters of the three links planar virtual linkage operated by the user in the simulations.

Link length (m)	**Link mass (kg)**	**Link inertia (kgm^2)**
$l_1 = 0.042$ m	$m_1 = 3$ kg	$I_1 = 0.015$ kgm^2
$l_2 = 0.042$ m	$m_2 = 3$ kg	$I_2 = 0.015$ kgm^2
$l_3 = 0.03$ m	$m_3 = 1$ kg	$I_3 = 0.005$ kgm^2



(a) Manipulation from third link.

(b) Manipulation from second link.

Figure 5.5: Simulated manipulations of a planar virtual linkage. Initial linkage position is shown in black. Linkage positions during manipulation are shown in grey.

The Simulink^{TM} diagram of the simulated manipulation is shown in Figure 5.6. In the diagram, the virtual environment is represented by the configuration space dynamics of the linkage. The haptic device controlled according to Equation (5.21) is represented by the operational space dynamics of the linkage. The haptic device controlled according to Equation (5.23) is represented by the operational space dynamics of the linkage augmented with the control signal:

$$\mathbf{u}_{add} = \mathbf{M}_d \sum_{i=1}^{n_{tc}} \left(k_{tc} \mathbf{n}_{tc,i}^T \left(\mathbf{x}_{h_{vt}} - \mathbf{x}_h \right) + b_{tc} \mathbf{n}_{tc,i}^T \left(\dot{\mathbf{x}}_{h_{vt}} - \dot{\mathbf{x}}_h \right) \right) \mathbf{n}_{tc,i}, \tag{5.24}$$

where $\mathbf{M}_d$ is considered constant throughout the workspace and equal to unity. The coordination between the device and the virtual tool has stiffness $\mathbf{K}_{pc} = (100 \text{ N/m} \quad 100 \text{ N/m} \quad 0.5 \text{ Nm/rad})^T$ and damping $\mathbf{B}_{pc} = (70 \text{ N/(m/s)} \quad 70 \text{ Nm/(rad/s)} \quad 0.375 \text{ N/(m/s)})^T$[5], while the stiffness and damping of the topological constraints are $k_{tc} = 10$ N/kg/m and $b_{tc} = 7$ N/kg/(m/s), respectively.

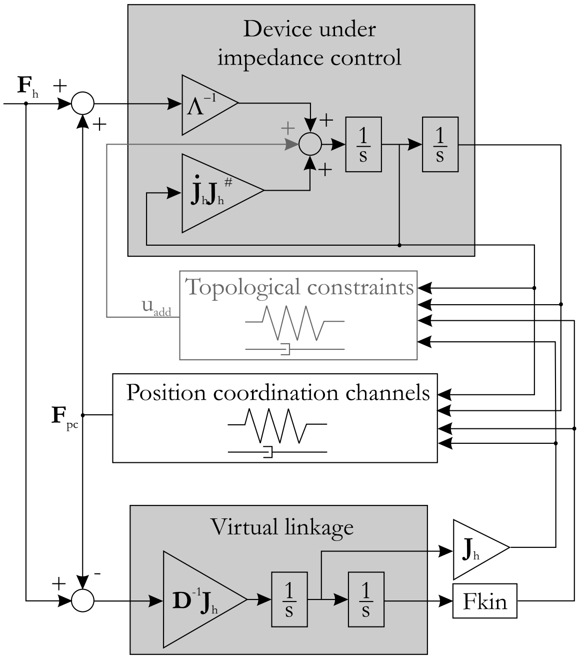
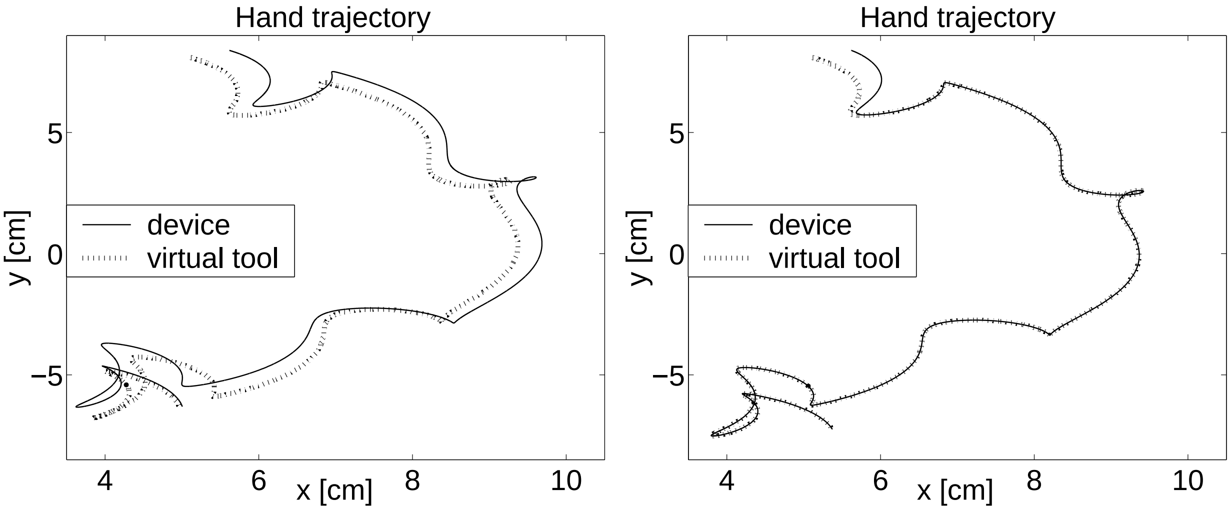


Figure 5.6: Simulink diagram of the haptic manipulation of a planar virtual linkage.

Figure 5.7 depicts the results for the first simulated interaction, in which the user manipulates the virtual tool from the COM of the distal link, as shown in Figure 5.5(a). The user's hand trajectories on the device and in the simulation are shown in Figure 5.7(a) for the case when topology is imposed at the acceleration level. They are shown in Figure 5.7(b) for the case when user's departure from the constraint manifold is penalized according to Equation (5.24). Since the user holds a link with 3 DOFs, the topological penalties are applied only intermittenly, when they user moves through a singularity. Note that the device drifts from the virtual tool when linkage constraints are imposed at the acceleration level. In contrast, the drift is substantially reduced through penalizing violations of the virtual topology.

The results for a second simulated interaction are shown in Figure 5.8. In this interaction, the user manip-

[5] They are chosen to match the values implemented on the haptic device used in the experiments [136].

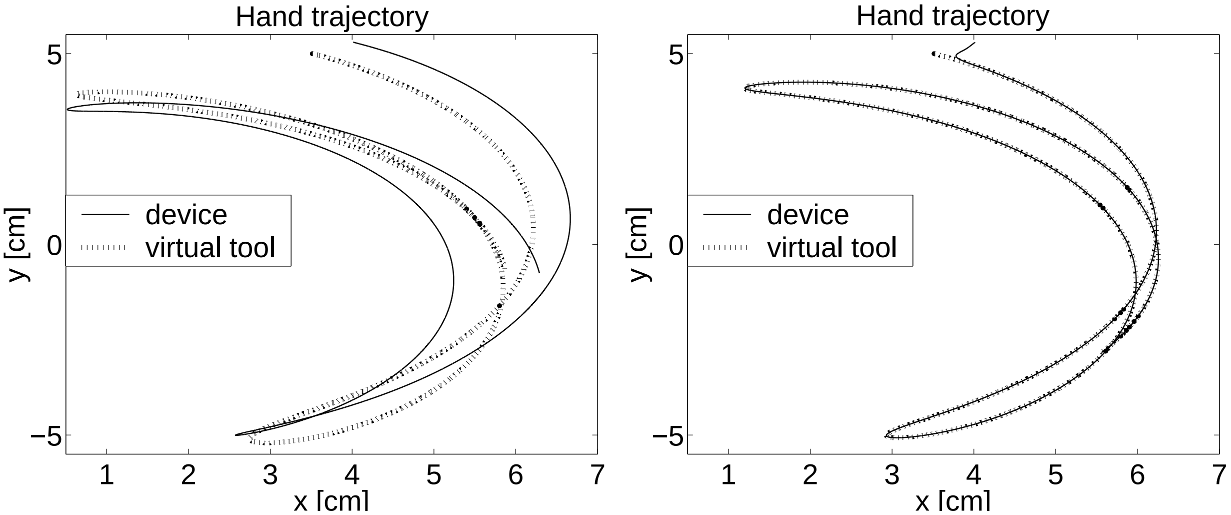


(a) Topology imposed at the acceleration level.

(b) Topology imposed through penalties.

Figure 5.7: Simulated planar manipulation of a three links virtual tool held from the COM of the distal link.

ulates the virtual tool from the COM of the middle link, as shown in Figure 5.5(b). This link has only 2 DOFs. Hence, the virtual topology constrains user's motion throughout the interaction. Once more, the trajectories in Figures 5.8(a) and 5.8(b) illustrate that the drift between the user and the virtual tool is significant unless user's departure from the configuration manifold is appropriately penalized through device control.



(a) Topology imposed at the acceleration level.

(b) Topology imposed through penalties.

Figure 5.8: Simulated planar manipulation of a three links virtual tool held from the COM of the second link.

5.4.2 Experiments

The control in Equation (5.23) imposes linkage topology on users through employing the virtual tool directions of topological constraint, $\mathbf{n}_{tc,i}$. In addition, the physically motivated mapping of the virtual dynamics from configuration space to operational space involves the computation of the null space of the linkage Jacobian at the user's hand, $\mathbf{J}_h$. Both the computation of the directions of topological constraint and the computation of the null space are based on the SVD of $\mathbf{J}_h$, which may be numerically expensive for the speed requirements of the haptic control loop. Suitable approximations of these directions, $\widehat{\mathbf{n}}_{tc,i}$, and of the virtual linkage dynamics are derived in the following chapter, in Section 6.3. Herein, they are used to illustrate the ability of the control in Equation (5.23) to limit users' motion as required by the virtual linkage that the users operate.

The controlled experiments presented in this section mimic linkage manipulation from various user-selected links. These experiments compare two types of trajectories. Trajectories obtained by enforcing topological constraints through penalizing the user's departure from the configuration manifold are contrasted to trajectories obtained by enforcing virtual topology through the inverse of the operational space inertia of the linkage (i.e., through controlling to zero user's acceleration along directions orthogonal to the configuration manifold).

In the experiments, a constant wrench $\mathbf{F}_h = (0.4\text{N} \quad 0\text{N} \quad 0\text{Nm})^T$ represents the user[6]. In the first experiment, the wrench is applied to the COM of the distal link of the linkage shown in Figure 5.9 (dimensional and inertial properties were given in Table 5.1). In the second experiment, the wrench is applied to the COM of the second link of the same linkage. In both experiments, the linkage is initially at rest, in the configuration space position shown in Figure 5.9 (i.e., $\mathbf{q}_0 = \left(0\text{rad} \quad \frac{\pi}{6}\text{rad} \quad -\frac{\pi}{3}\text{rad}\right)^T$). Asynchronous communication is used. The linkage dynamics are approximated by:

$$\widehat{\mathbf{D}}_{vt}\ddot{\mathbf{q}} = \mathbf{J}_h^T \mathbf{F}_h, \tag{5.25}$$

i.e., the linkage moves freely under the controlled hand wrench. In Equation (5.25), $\widehat{\mathbf{D}}_{vt}$ is the approximate inertia matrix of the virtual tool and is computed as described in Section 6.3.1. Furthermore, the control in Equation (5.24) is approximated by:

$$\mathbf{u}_{add} = \mathbf{M}_d \sum_{i=1}^{n_{tc}} \left(k_{tc}\widehat{\mathbf{n}}_{tc,i}^T \left(\mathbf{x}_{h_{vt}} - \mathbf{x}_h\right) + b_{tc}\widehat{\mathbf{n}}_{tc,i}^T \left(\dot{\mathbf{x}}_{h_{vt}} - \dot{\mathbf{x}}_h\right)\right) \widehat{\mathbf{n}}_{tc,i}, \tag{5.26}$$

where $\widehat{\mathbf{n}}_{tc,i}$ approximates i-th topological constraint direction, as described in the following chapter, in Section 6.3.2. The gains of the control in Equation (5.26) are $k_{tc} = 200\text{N}/(\text{kg}\cdot\text{m})$ and $b_{tc} = 50\text{N}/(\text{kg}\cdot(\text{m/s}))$.

The user's hand trajectories on the haptic device ("HD") and in the simulated virtual environment ("VE") plotted in Figure 5.10 correspond to manipulation from the distal link. The trajectories shown in Figure 5.11 represent manipulation from the second link. Figures 5.10(a) and 5.11(a) demonstrate that the control in Equation (5.24) effectively imposes the topological constraints on users regardless of the link they operate. The user's hand trajectory on the device follows the simulated user's hand trajectory in the virtual environment

[6] For the impedance type haptic interface employed in these experiments, a constant wrench represents a worst case approximation for stability because it eliminates the adaptive damping of the user's hand [64].

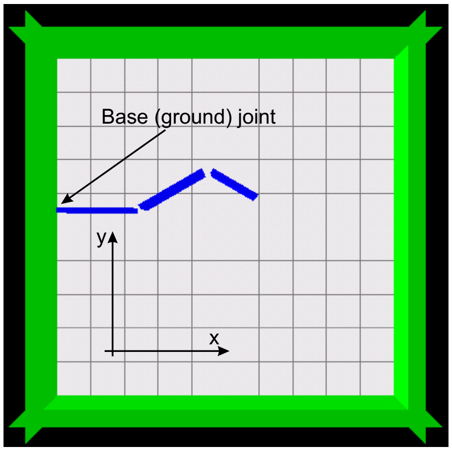
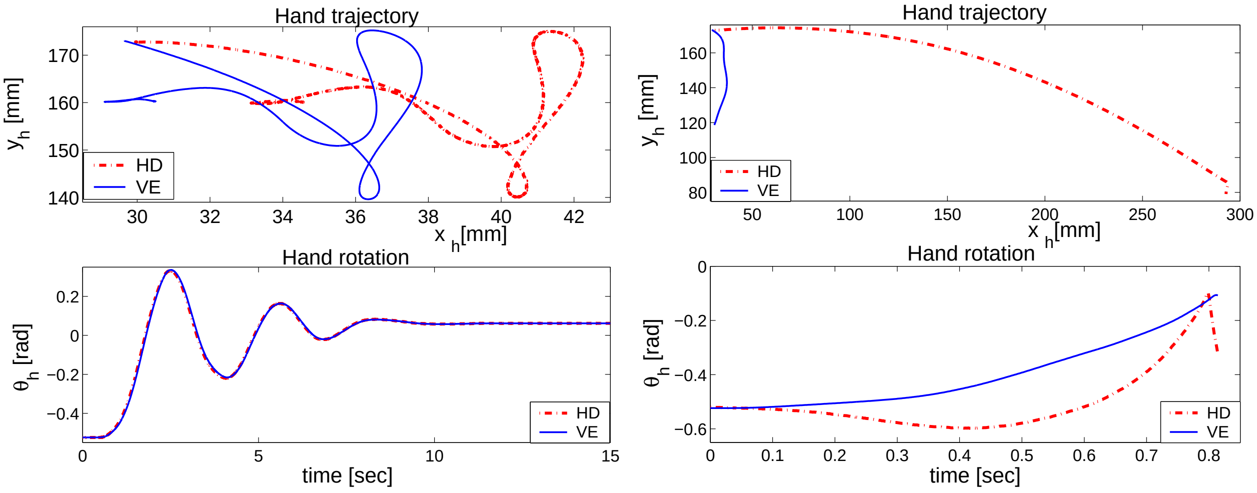


Figure 5.9: Testbed virtual environment for controlled linkage manipulation from any user-selected link.

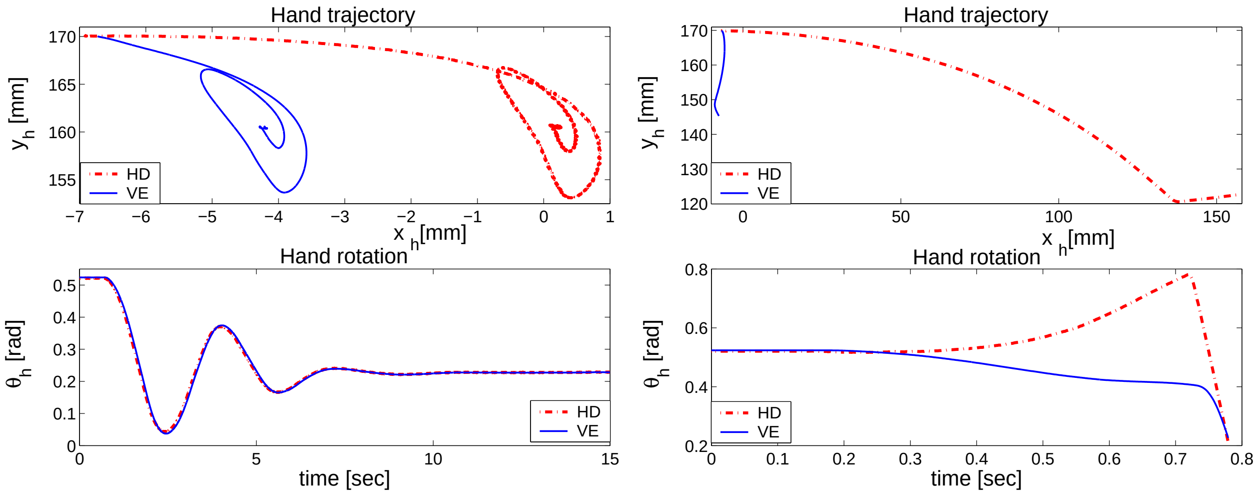
within the steady state error due to the limited stiffness of the penalties applied to users. On the other hand, Figures 5.10(b) and 5.11(b) show that the drift between the user's hand trajectories on the haptic device and in the virtual environment simulation is large when virtual topology is enforced only at the acceleration level. The impedance controller alone cannot enforce the virtual topology. The user's hand freely moves away from the configuration manifold until it reaches the mechanical boundary of the workspace of the haptic interface. In Figures 5.10(b) and 5.11(b), the sudden changes occuring in the trajectories of the user's hand on the haptic device after approximately 0.7s represent hard constraints due to device workspace limitations.



(a) Topology enforced through penalties.

(b) Topology enforced at acceleration level.

Figure 5.10: Haptic manipulation from the distal link of the virtual linkage shown in Figure 5.9. Users apply a constant wrench $\mathbf{F}_h = (0.4\text{N} \quad 0\text{N} \quad 0\text{Nm})^T$.



(a) Topology enforced through penalties.

(b) Topology enforced at acceleration level.

Figure 5.11: Haptic manipulation from the second link of the virtual linkage shown in Figure 5.9. Users apply a constant wrench $\mathbf{F}_h = (0.4\text{N} \quad 0\text{N} \quad 0\text{Nm})^T$.

The experiments described in this section have demonstrated realistic haptic manipulation of serial linkages from various user-selected links. In these experiments, motion constraints due to linkage topology have been imposed on users through penalties applied along the linkage directions of infinite structural stiffness (i.e., orthogonal to the linkage configuration manifold). A virtual coupler controller [3,42] would also allow users to perceive the linkage topology. However, the control proposed in this thesis allows larger gains to be used for enforcing the virtual topology. This is because the proposed control is applied only when users depart from the configuration manifold and its gains are limited only by the device itself and by the sampling rate. On the other hand, the virtual coupler acts throughout the interaction and its gains must be chosen so as to achieve a compromise between free and constrained motion. While large gains are necessary during constrained motion, low gains are desirable during free motion. Since the proposed control enforces the linkage topology through larger gains than the virtual coupler, it enables users to perceive stiffer topological constraints. Hence, it allows more transparent interaction.

This chapter has proposed simulation and control techniques that enable convincing and unrestricted haptic manipulation of virtual linkages from any user-selected link. Physically motivated mappings of the virtual dynamics from configuration space to operational space have been derived. A control penalizing users' departure from the configuration manifold of the linkage has been proposed for limiting users' motion as required by the virtual linkage that they operate. Simulations and experiments performed using a planar haptic interaction system have demonstrated that the techniques described in this section can be used to enable unrestricted operation of virtual linkages. In the experiments, the real time performance of these techniques has been guaranteed through efficiently implementating them in a local model of interaction. This implementation is detailed in the following chapter.

Chapter 6

Efficient haptic rendering of rigid body motion with constraints

This chapter presents a local model of interaction that is suitable for adding realistic forces and torques to the manipulation of virtual tools within rigid multibody virtual environments. The local model efficiently implements the earlier developed simulation and control techniques that facilitate transparent haptic rigid body interaction within virtual worlds. After presenting a model synopsis, the chapter details the geometric and the dynamic approximations that enable the proposed local model to achieve haptic speed requirements. The chapter ends with an experimental comparison between the proposed local model of rigid body interaction and prior local models.

The schematic of haptic virtual tool manipulation using the proposed local model is shown in Figure 6.1. In this figure, dashed lines indicate low bandwidth, asynchronous communication between the virtual environment and the local model of interaction. Solid lines indicate high bandwidth, synchronous communication between the local model and the device. For simplicity, position coordination signals are not represented.

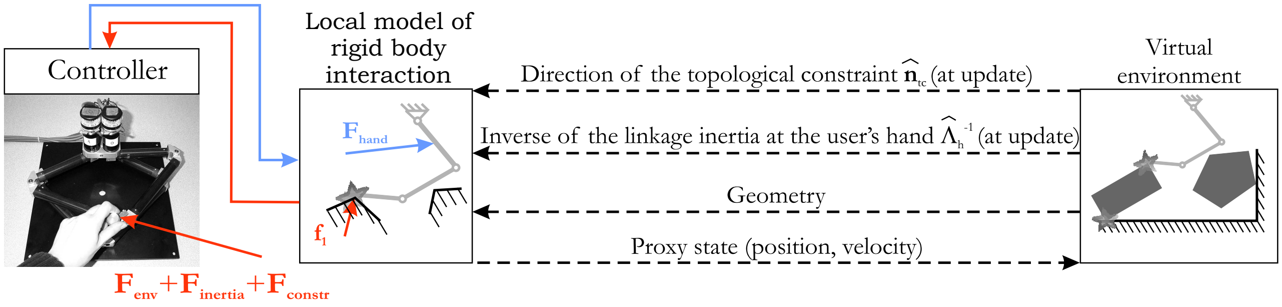

Controller
Local model of rigid body interaction
Virtual environment
Direction of the topological constraint $\widehat{\mathbf{n}}_{tc}$ (at update)
$\mathbf{F}_{hand}$
Inverse of the linkage inertia at the user's hand $\widehat{\Lambda}_h^{-1}$ (at update)
Geometry
$\mathbf{f}_1$
Proxy state (position, velocity)
$\mathbf{F}_{env}+\mathbf{F}_{inertia}+\mathbf{F}_{constr}$

Figure 6.1: Realistic haptic manipulation of virtual tools using the local model of interaction proposed in this thesis.

6.1 Synopsis of the local model

To meet the need of haptic devices for physically motivated forces and torques provided at high fixed frequencies, the local model of interaction implements a less complex simulation. The local model approximates the interaction between the virtual tool and the virtual environment through the interaction between the virtual tool and nearby objects only (see Figure 6.2). A proxy of the virtual tool and constraints imposed on the motion of the virtual tool by nearby objects comprise the model. The quality of the approximation is maintained by updating the local model at each step of the virtual environment simulation. Coordination between the proxy and the haptic device is achieved using force and state (i.e., position and velocity) information, while coordination between the proxy and the virtual tool in the virtual environment is achieved using proxy state (i.e., position and velocity) information.

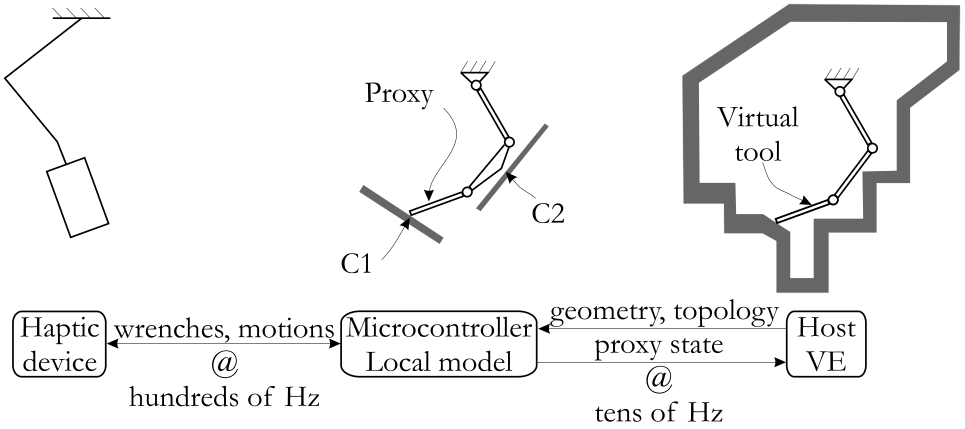


Figure 6.2: Communication between the virtual environment (VE), the local model of interaction, and the haptic device.

Four features distinguish this local model of interaction from existing models. First, the model considers not only the virtual objects already in contact with the virtual tool, but also nearby virtual objects. As shown in Section 6.2.3, prediction of virtual tool contacts with nearby virtual objects effectively alleviates the delay in updating local geometry when users operate the virtual tool through small clearances. Second, the local model computes impulsive forces upon contact in addition to penalty and friction forces during contact. By dissipating the kinetic energy of the user's hand as prescribed by the coefficient of restitution and allowed by the device capabilites, the impulsive forces increase the stability of the virtual contacts. By enforcing user's hand accelerations that are much larger upon contact than the user's hand accelerations imposed by the penalty forces during contact, the impulsive forces enhance the perceived rigidity of the virtual contacts. Third, the model includes a dynamic proxy of the virtual tool in addition to local geometry. The dynamic proxy allows a four channel haptic controller [136] to be used that both transmits wrenches between the device and the proxy and coordinates positions between them. Compared to the virtual coupler [3, 30], the four channel controller transmits the impulsive wrenches computed by the local model of interaction to the user's hand. Hence, it enables users to feel collisions upon impact and thereby increases the perceived rigidity of the virtual objects [90, 125]. Compared to directly coupling an impedance device to an admittance simulation [21], the four channel controller allows stiffer contacts to be stably rendered to users. This is demonstrated through experiments in Section 6.4.3. Lastly, the new local model can be used to add realistic haptic feedback to interactive virtual environments generated using any commericial simulation package. This is because the model makes no assumptions about the data structures or the collision detection and the dynamic response algorithms used by the virtual environment simulation. This feature is illustrated in Section 6.4 by using the proposed local model of interaction to allow haptic interaction within a testbed virtual world generated using Vortex^{TM}, a physics engine developed by CMLabs Simulations Inc (www.criticalmasslabs.com).

The features of the local model of interaction are presented in the following sections, starting with the local geometry in Section 6.2. The local dynamics are detailed in Section 6.3.

6.2 Local geometry

The forces and torques that provide a convincing haptic experience arise at the contacts between the virtual tool and other virtual objects, collectively referred to as the virtual environment. Hence, the local representation of the virtual world geometry is important for realistic force feedback. Complete geometry is desirable for accurate computation of interactions, since it allows all contact transitions of the virtual tool to be resolved locally. Nevertheless, complete geometry cannot be used in the local model due to the complexity and the variability across simulation packages of the data structures and of the collision detection algorithms associated with complex geometric models.

Partial (i.e., some subset of the) virtual environment geometry may result in delayed updating of the virtual tool contacts in the local model of the interaction. In turn, this computational delay may cause perceptual artifacts or unstable interaction due to undesirable force discontinuities at model updates. An example interaction during which undesirable force discontinuities arise at model updates is depicted in Figure 6.3. Successive time steps of the virtual environment are shown in Figures 6.3(a), 6.3(b), and 6.3(c), and the corresponding local model updates are shown in Figures 6.3(d), 6.3(e), and 6.3(f). The example illustrates the case where only virtual environment geometry in contact with the rectangular virtual tool is sent to the local model. Note that the virtual tool contacts arrive late, i.e., with significant penetration, to the local model and cause force discontinuities at updates. In turn, these force discontinuities may destabilize the interaction.

The selection of salient geometry to be used in the local model and the choice of a method for transitioning between the old and the new local models are critical for a stable and convincing virtual kinesthetic experience. The techniques employed for selecting appropriate local geometry and for transitioning between successive local models are presented in the following subsection.

6.2.1 Active geometry

While the geometry of the virtual world is important for convincing haptic feedback, its importance varies temporally and spatially. Thus, the instantaneous interactions of the virtual tool are independent of the

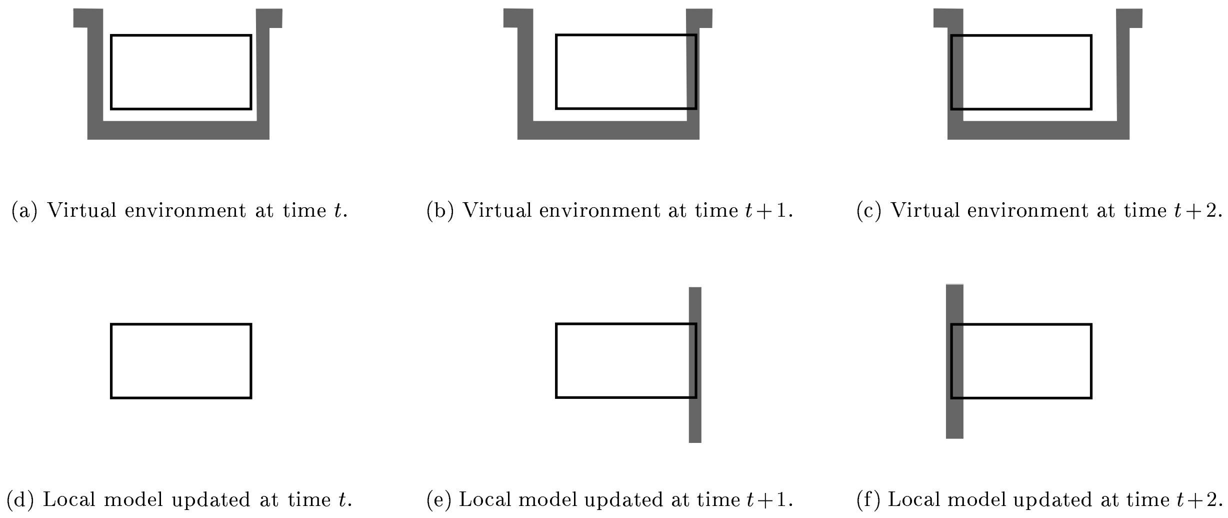

(a) Virtual environment at time t.
(b) Virtual environment at time $t+1$.
(c) Virtual environment at time $t+2$.
(d) Local model updated at time t.
(e) Local model updated at time $t+1$.
(f) Local model updated at time $t+2$.

Figure 6.3: Example haptic interaction during which the computational delay of the virtual environment results in delayed updating of the virtual tool contacts in the local model of interaction.

geometry of virtual objects that are not currently in the same contact group with the virtual tool (i.e., in contact with it either directly or through other objects). For example, the geometry of body C in Figure 6.4 has no influence on the forces presently acting on the virtual tool. Furthermore, the virtual tool interactions are influenced more by the geometry of virtual objects in direct contact with the virtual tool than by the geometry of virtual objects in indirect contact with it. For example, consider that a force $\mathbf{f}_A$ is applied to A and a force $\mathbf{f}_B$ is applied to B in Figure 6.4. Both these forces are transmitted to the virtual tool via the contact between the virtual tool and A. Hence, the magnitude of the force applied to the virtual tool depends both on $\mathbf{f}_A$ and on $\mathbf{f}_B$. However, the direction of the interaction force between the virtual tool and the virtual environment depends on the geometry of A and is independent of the geometry of B. Thus, virtual environment geometry in direct contact with the virtual tool at simulation updates, called active geometry, is of primary importance for realistic user interaction within the virtual world. The active geometry is chosen initially to approximate the virtual environment in the local model of the interaction.

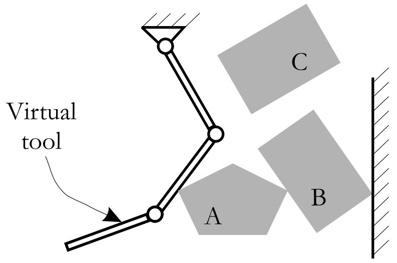


Figure 6.4: The influence of the geometry of the virtual environment on the interactions of the virtual tool.

Coupling of a device to an arbitrary virtual world is enabled by restricting the active geometry to information generally available in physically-based virtual environments. Typical simulation packages represent rigid bodies as collections of convex polyhedra and compute contacts between pairs of these polyhedra [85]. They approximate rigid body contact through a finite number of vertex-face and/or edge-edge contacts [152]. For each contact, they provide a contact point $\underset{\sim}{\mathbf{c}}_{p,VE}$, a constraint normal direction $\mathbf{n}$, and a penetration depth. The constraint normal direction is the direction of motion of the virtual tool contact point that is disallowed by the contact. The penetration depth is equal and opposite to the separation distance s between the bodies; i.e., it is positive when the virtual tool and the virtual environment overlap. The *active geometry* consists of all contacts of the virtual tool with the virtual environment.

In the local model of interaction, the active geometry is encapsulated in the local proxy constraints. A *local proxy constraint* is defined through the identifiers of the two contacting objects and the constraint geometric, kinematic, and dynamic properties. Typically, the identifiers of the contacting objects are the addresses of their data structures in the simulation. The geometric properties of a local proxy constraint embed the geometric information provided by typical simulation packages for each contact of the virtual tool (see Figure 6.5). As depicted in Figure 6.5(d), these properties consist of:

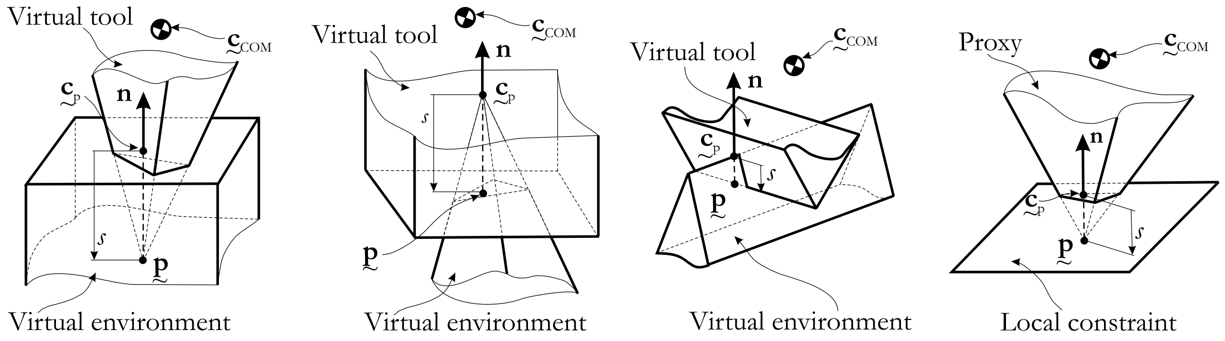
- the *local constraint position* $\underset{\sim}{\mathbf{c}}_p$. This is the position of the face point closest to the vertex when the virtual tool is in vertex-face contact with the virtual environment (Figure 6.5(a)); it is the position of the vertex when the virtual tool is in face-vertex contact with the virtual environment (Figure 6.5(b)); and, it is the position of the point on the environment edge that is closest to the virtual tool edge when the virtual tool is in edge-edge contact with the virtual environment (Figure 6.5(c)).
- the *local constraint normal direction* $\mathbf{n}$. This is the constraint normal direction supplied by the simulation.
- the *local contact point* $\underset{\sim}{\mathbf{p}}$. This is the vertex itself when the virtual tool is in vertex-face contact with the virtual environment (Figure 6.5(a)); it is the face point closest to the vertex when the virtual tool is in face-vertex contact with the virtual environment (Figure 6.5(b)); and, it is the point on the virtual tool edge closest to the environment edge when the virtual tool is in edge-edge contact with the virtual environment (Figure 6.5(c)).

Note that constraints are considered to extend infinitely in the local model of interaction. Moreover, a proxy-centric view is adopted in describing the active geometry: constraint normals are directed from the virtual environment to the virtual tool and the contact point is taken on the proxy.

It is assumed that both the local constraint position $\underset{\sim}{\mathbf{c}}_p$ and the local constraint normal direction $\mathbf{n}$ are provided by the virtual environment simulation in world coordinates[1]. The local contact point $\underset{\sim}{\mathbf{p}}$ is in proxy coordinates and is computed locally by:

$$\underset{\sim}{\mathbf{p}} = \mathbf{R}\left(\underset{\sim}{\mathbf{c}}_p + s\mathbf{n} - \underset{\sim}{\mathbf{c}}_{COM}\right). \tag{6.1}$$

[1] If this is not the case, they can be transformed into world coordinates through an additional rigid body transformation and an additional rotation, respectively.


Virtual tool
$\underset{\sim}{\mathbf{c}}_{COM}$
$\underset{\sim}{\mathbf{c}}_P$
n
s
$\underset{\sim}{\mathbf{p}}$
Virtual environment
Virtual tool
$\underset{\sim}{\mathbf{c}}_{COM}$
n
$\underset{\sim}{\mathbf{c}}_P$
s
$\underset{\sim}{\mathbf{p}}$
Virtual environment
Virtual tool
$\underset{\sim}{\mathbf{c}}_{COM}$
n
$\underset{\sim}{\mathbf{c}}_P$
s
$\underset{\sim}{\mathbf{p}}$
Virtual environment
Proxy
$\underset{\sim}{\mathbf{c}}_{COM}$
n
$\underset{\sim}{\mathbf{c}}_P$
s
$\underset{\sim}{\mathbf{p}}$
Local constraint

(a) Virtual tool in vertex-face contact with the virtual environment.

(b) Virtual tool in face-vertex contact with the virtual environment.

(c) Virtual tool in edge-edge contact with the virtual environment.

(d) Local contact.

Figure 6.5: Geometric information provided by typical simulation packages for each type of virtual tool contacts (Figures 6.5(a), 6.5(b), and 6.5(c)) and its representation in the local contact geometry (Figure 6.5(d)).

In Equation (6.1), $\mathbf{R}$ is the rotation from the world to the proxy coordinates and $\underset{\sim}{\mathbf{c}}_{COM}$ is the position of the COM of the proxy. The kinematic properties of a local proxy constraint are provided by the local constraint velocity v_{constr}, which is the projection along the constraint normal $\mathbf{n}$ of the velocity of the virtual environment point in contact with the virtual tool $\mathbf{v}_{\underset{\sim}{\mathbf{c}}_P}$:

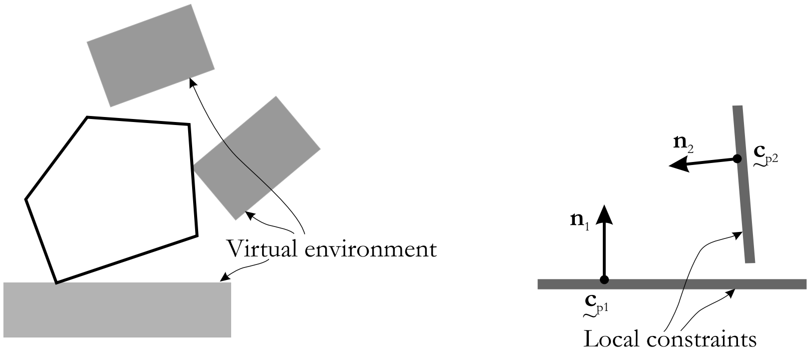
$$v_{constr} = \mathbf{n}^T \mathbf{v}_{\underset{\sim}{\mathbf{c}}_P} . \tag{6.2}$$

The local constraint velocity is used to predict the constraint position at the next simulation update, thus diminishing local geometry discontinuities at model updates. The dynamic properties of a local proxy constraint consist of the constraint stiffness $K_{contact}$, the constraint damping $B_{contact}$, the coefficient of restitution e, and the coefficients of static and kinetic friction, μ_s and μ_k, respectively. While all local proxy constraints have the same stiffness, damping, and coefficient of restitution, they may have different coefficients of friction.

The local proxy constraints are the only geometry that the local model is aware of. Hence, only partial virtual tool and virtual environment geometry is available locally. This is illustrated in Figure 6.6, where the local representation of an example virtual world geometry is depicted. Partial geometry makes the proposed approach compatible with simulation packages regardless of the data structures that they use for representing the virtual objects. Moreover, it simplifies local collision detection, which becomes an iteration through all local constraints in order to compute separation distances according to:

$$s = \mathbf{n}^T \left(\underset{\sim}{\mathbf{c}}_{COM} + \mathbf{R}^{-1} \underset{\sim}{\mathbf{p}} - \underset{\sim}{\mathbf{c}}_p \right) . \tag{6.3}$$

In Equation (6.3), s is the separation between the proxy and the local constraint of interest and $\mathbf{R}^{-1}$ is the rotation from proxy to world coordinates.


Virtual environment
$\mathbf{n}_2$
$\underset{\sim}{\mathbf{c}}_{p2}$
$\mathbf{n}_1$
$\underset{\sim}{\mathbf{c}}_{p1}$
Local constraints

(a) Example contact geometry in the virtual environment simulation.

(b) Local representation of the virtual environment geometry shown in Figure 6.6(a).

Figure 6.6: Partial virtual environment geometry available in the proposed local model of interaction.

The active geometry shifts the computational delay of the simulation from delay in computing interaction forces to delay in updating local geometry. As a result of faster force computation, the users' haptic experience is improved in two ways: (i) they can manipulate much stiffer objects, due to the high control rate that can be achieved; and (ii), they can feel physical phenomena that rely on fast force transitions, such as collisions and stick-slip friction. However, at model updates, undesirable discontinuities may arise in the locally computed forces that may destabilize the interaction. Two techniques are proposed to avoid such discontinuities: local proxy deformation and the use of ϵ-active local geometry. These techniques are detailed in the following Sections 6.2.2 and 6.2.3.

6.2.2 Local proxy deformation

An update of the local model may result in: (i) significant proxy penetration into new local constraints (this may happen when the user moves the virtual tool very quickly into a constraint); and (ii) existing local constraints with significant discontinuities in the penetration depth (this may happen when the virtual environment is generated using a penalty-based simulation, because the proxy interactions are approximations of the virtual tool interactions and the proxy penetration into the local constraints may differ from the virtual tool penetration into the virtual environment). Discontinuities in the local constraint penetration lead to discontinuities in the setpoint of the force control loop and may produce unacceptable force spikes or may destabilize the interaction. This difficulty is addressed by maintaining the proxy penetration into the local constraints continuous through proxy , i.e., the new local contact point is computed by (see Figure 6.7):

$$\underset{\sim}{\mathbf{p}} = \mathbf{R}\left(\underset{\sim}{\mathbf{c}}_{p} - \underset{\sim}{\mathbf{c}}_{COM}\right). \tag{6.4}$$

When the virtual environment sends an existing constraint to the local model, the local contact point is recomputed by:

$$\underset{\sim}{\mathbf{p}} = \mathbf{R}\left(\underset{\sim}{\mathbf{c}}_p + s\mathbf{n} - \underset{\sim}{\mathbf{c}}_{COM}\right). \tag{6.5}$$

The proxy is expanded back towards the actual geometry of the virtual tool whenever a local constraint becomes inactive. This approach is independent of the stiffness of the local constraints.

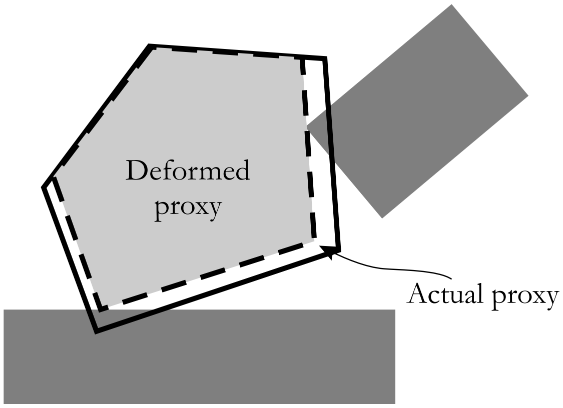


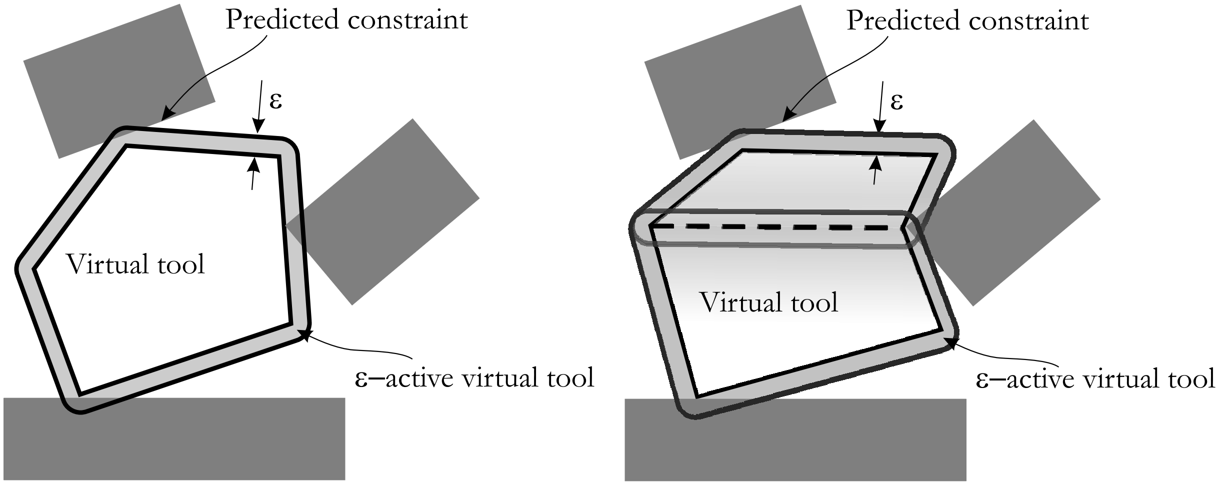
Figure 6.7: Local proxy deformation due to violation of new constraints.

Local proxy deformation maintains penetration continuity and eliminates force discontinuities at model updates. However, it allows the proxy to drift relative to the virtual tool. This drift is noticeable during fast motions in locally cluttered virtual environments, when significant proxy deformation may occur in order to maintain penetration continuity. Therefore, the free space perceived haptically may be much larger than the free space perceived visually. ϵ-active local geometry is used in the proposed local model to alleviate drift.

6.2.3 ϵ-active local geometry

To diminish drift and allow realistic haptic rendering of small clearances (in virtual environments generated using constraint-based simulations) and of tight constraints (in virtual worlds generated using penalty-based simulations), the local model is augmented by including constraints within some neighborhood of the virtual tool. The relevant neighborhood is defined by sweeping a sphere of radius ϵ over the volume of the virtual tool, as described in [88] and depicted in Figure 6.8(a). If the virtual tool has concave surfaces, then typical simulation packages represent it as a collection of convex polyhedra [59, 85] and the augmented virtual tool (hereafter called ϵ-active virtual tool) is obtained by sweeping a sphere of radius ϵ over the volume of each component polyhedron, as illustrated in Figure 6.8(b).

This technique reduces the drift by adding prediction capabilities to the proposed model: constraints are sent to the local model before they become active. Moreover, the approach is simple to implement and general enough to be applicable to any virtual environment. In addition to diminishing drift, the ϵ-active virtual tool selects a unique constraint normal at degenerate contacts, as illustrated in Figure 6.9 for a vertex-vertex contact



(a) Convex ϵ-active virtual tool.

(b) Concave ϵ-active virtual tool.

Figure 6.8: Neighboring constraints included in the local model through the use of an ϵ-active virtual tool, obtained by sweeping a sphere of radius ϵ over the volume of the virtual tool.

of the virtual tool with the virtual environments. Hence, it alleviates the singularity in the constraint normal computation at degenerate contacts.

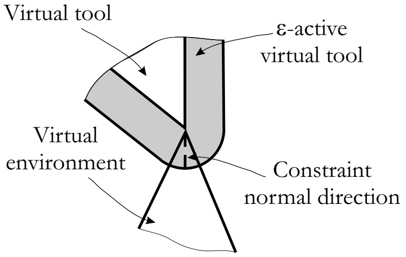


Figure 6.9: The ϵ-active virtual tool eliminates the singularity in the constraint normal computation at a vertex-vertex contact between the virtual tool and the virtual environment by selecting a unique constraint normal direction.

At simulation updates, ϵ-active geometry is included in the local model by computing the local contact position for new local constraints according to:

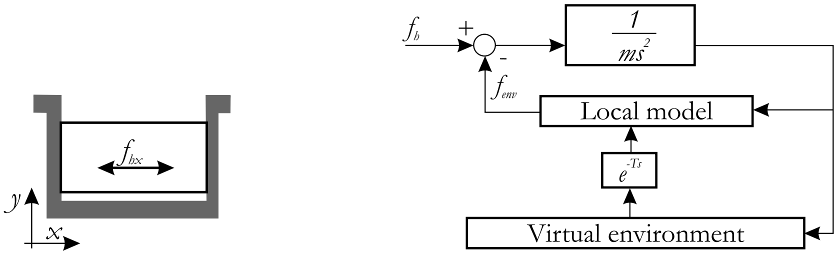
$$\underset{\sim}{\mathbf{p}} = \mathbf{R}\left(\underset{\sim}{\mathbf{c}}_p + (s-\epsilon)\,\mathbf{n} - \underset{\sim}{\mathbf{c}}_{COM}\right), \tag{6.6}$$

and by updating the local contact position for existing constraints according to:

$$\begin{cases} \underset{\sim}{\mathbf{p}} = \mathbf{R}\left(\underset{\sim}{\mathbf{c}}_p + (s-\epsilon)\,\mathbf{n} - \underset{\sim}{\mathbf{c}}_{COM}\right) & \text{if } s_p < s-\epsilon < 0 \text{ or } 0 < s-\epsilon \\ \underset{\sim}{\mathbf{p}} = \mathbf{R}\left(\underset{\sim}{\mathbf{c}}_p + s\mathbf{n} - \underset{\sim}{\mathbf{c}}_{COM}\right) & \text{if } s-\epsilon < s_p < 0 \\ \underset{\sim}{\mathbf{p}} = \mathbf{R}\left(\underset{\sim}{\mathbf{c}}_p - \underset{\sim}{\mathbf{c}}_{COM}\right) & \text{if } s-\epsilon < 0 < s_p \end{cases} \tag{6.7}$$

In Equations (6.6) and (6.7), s is the separation between the ϵ-active virtual tool and the virtual environment as reported by the simulation, and s_p is the separation between the proxy and the local constraint in the local model of the interaction.

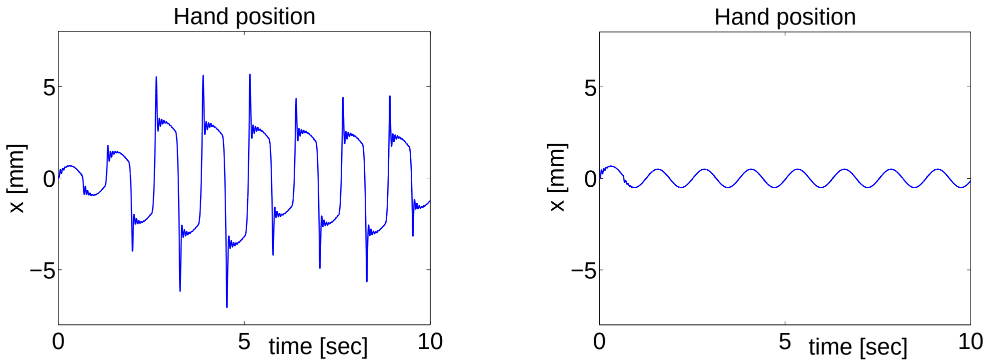
The ability of the ϵ-active geometry to improve user's perception of tightly constrained virtual tools is illustrated through a Matlab^{TM} simulation of a planar (3 DOF) peg-in-hole manipulation. In this simulation, the commercial package generating the virtual environment is implemented as a penalty-based simulation of a 1kg peg in a tight hole that fits the peg exactly (see Figure 6.10(a)). The computational delay of the package is chosen 20ms. Furthermore, the user's action is implemented as a horizontal force $f_{h_x} = 5sin\left(\frac{2\pi}{10}\right)$N. In the simulation, the effect of the ϵ-active geometry is illustrated separately from the effect of the impulsive forces and the effect of the haptic controller. The effect of the impulsive forces is eliminated by implementing penalty-based local constraints (with stiffness $K_{contact}$ = 10000N/m and damping $B_{contact}$ = 15N/(m/s)). As depicted in Figure 6.10(b), the effect of the haptic controller is eliminated by directly coupling the local model to the haptic device (i.e., by directly applying the environment forces to the user's hand). The trajectory of the user's hand is plotted in Figure 6.11(a) for the case where only active geometry is included in the local model, and in Figure 6.11(b) for the case where ϵ-active geometry within a 5mm neighborhood of the virtual tool is sent to the local model. As shown in the plots, the drift is significant without ϵ-active geometry and is eliminated when an ϵ-active virtual tool is employed. Note that the motion of the user's hand is due only to the limited stiffness of the local constraints in the latter case.



(a) Virtual task.

(b) Dynamic model of the task.

Figure 6.10: One dimensional peg-in-hole task used to illustrate the influence of local geometry on user's perception of locally cluttered virtual environments.



(a) Active geometry.

(b) ϵ-active geometry (within 5mm of the virtual tool).

Figure 6.11: Hand position for the simulated peg-in-hole task depicted in Figure 6.10(a).

6.2.4 Contact clustering and constraint coherence

Local proxy deformation and an ϵ-active virtual tool smooth forces at model updates and improve the perception of tight virtual spaces provided that constraint continuity is ensured at these updates. Constraint continuity requires individual constraints to be identified and their local geometry to be updated according to Equations (6.5) and (6.6). In the local model of interaction, constraint continuity is maintained through contact clustering [85] and through maintaining constraint coherence based on geometric coherence [38].

Contact clustering is necessary in order to improve the stability of the haptic interaction in nonconvex regions of penalty-based virtual environments [85]. Typical collision detection algorithms employed in interactive simulations decompose nonconvex polyhedra into collections of convex pieces and approximate the penetration depths between pairs of overlapping convex pieces using algorithms such as proposed in [33, 59, 83, 147]. In nonconvex regions of the virtual world, the convex decomposition of the virtual objects may result in a large number of convex pieces around concavities. In turn, this may lead to a large number of contacts and, hence, a larger resultant environment stiffness which may induce instability in certain configurations [85]. Contact clustering aims to avoid instability caused by large environment stiffnesses arising from the large number of contacts reported by the collision detection algorithms in concave regions of the virtual world.

Contact clustering is performed in the virtual environment simulation using a translational proximity threshold. Specifically, a cluster is a collection of n_{cc} contacts of the virtual tool that satisfy the condition that their contact point $\underset{\sim}{\mathbf{c}}_{p,i}$ is within the translational threshold $\epsilon_{t_{VE}}$ from the contact points $\underset{\sim}{\mathbf{c}}_{p,j}$ of all other contacts in the cluster:

$$|\underset{\sim}{\mathbf{c}}_{p,i} - \underset{\sim}{\mathbf{c}}_{p,j}| \leq \epsilon_{t_{VE}} \quad \forall j = 1 \ldots n_{cc}. \tag{6.8}$$

A cluster of virtual tool contacts in the virtual environment is represented by one local contact (see Figure 6.12).

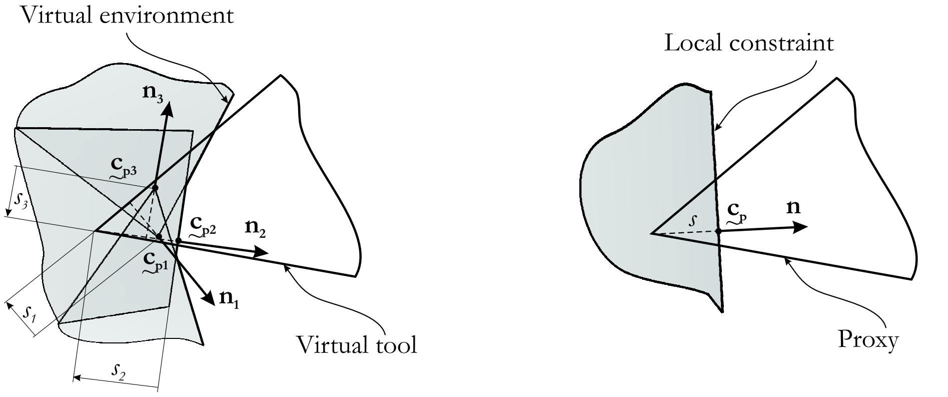
The separation distance of the local contact is computed by:

$$s = \min_j s_j \quad \forall j = 1 \ldots n_{cc}, \tag{6.9}$$

thus ensuring that the local contact exists as long as there are contacts in the cluster. Furthermore, the normal of the local contact is obtained averaging the normals of all cluster contacts:

$$\mathbf{n} = \frac{\sum_{j=1}^{n_{cc}} \mathbf{n}_j}{|\sum_{j=1}^{n_{cc}} \mathbf{n}_j|}. \tag{6.10}$$

Note that the translational threshold needs to be chosen based on the stiffness of the contacts in the simulation. The more compliant the virtual environment, the larger this threshold needs to be.


Virtual environment
$\mathbf{n}_3$
$\underset{\sim}{\mathbf{c}}_{p3}$
s_3
$\underset{\sim}{\mathbf{c}}_{p2}$
$\mathbf{n}_2$
$\underset{\sim}{\mathbf{c}}_{p1}$
$\mathbf{n}_1$
s_1
s_2
Virtual tool
Local constraint
s
$\underset{\sim}{\mathbf{c}}_{p}$
$\mathbf{n}$
Proxy

(a) Cluster of contacts in the virtual environment.

(b) Local constraint representing the cluster in Figure 6.12(a).

Figure 6.12: Contact clustering in the virtual environment.

Besides contact clustering in the virtual environment, geometric coherence [38] is used in the local model to maintain the temporal coherence of the local constraints at model updates[2]. In other words, constraints existing at two consecutive updates, $t-1$ and t, are considered the same local constraint if their contact points and normals are within some translational $\epsilon_{t_{LM}}$ and rotational $\epsilon_{r_{LM}}$ proximity thresholds, respectively, from each other:

$$\begin{aligned} |\underset{\sim}{\mathbf{c}}_{p_t} - \underset{\sim}{\mathbf{c}}_{p_{t-1}}| &\leq \epsilon_{t_{LM}}, \qquad (6.11)\\ \mathbf{n}_t^T \mathbf{n}_{t-1} &\geq \epsilon_{r_{LM}}. \qquad (6.12)\end{aligned}$$

Time coherence breaks down if the virtual objects move more than prescribed by $\epsilon_{t_{LM}}$ and $\epsilon_{r_{LM}}$ during one

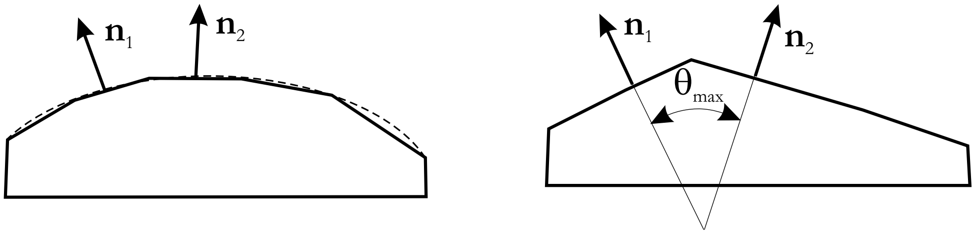
[2] This is necessary because the geometry of the local constraints does not include connectivity information. Lack of connectivity ensures that less information is sent to the local model, that local collision detection is computationally inexpensive, and that the local model is compatible with any commercial simulation package.

simulation step. Therefore, these thresholds must be chosen based on the maximum speed of the objects in the virtual environment and the average time step of the simulation, Δt_{avg}. The translational proximity threshold is computed by:

$$\epsilon_{t_{LM}} = \Delta t_{avg} v_{max}, \tag{6.13}$$

where the maximum speed of the virtual objects v_{max} is set by the user in the virtual environment and Δt_{avg} is computed in the local model by averaging the last five time steps of the simulation. Similarly, the rotational proximity threshold $\epsilon_{r_{LM}}$ is computed using the maximum angular speed of the virtual objects ω_{max} set by the user in the virtual world. In the local model, the rotational proximity threshold is also used to distinguish between discrete approximations of round virtual objects through polyhedra and polyhedral virtual objects (see Figure 6.13). Therefore, the rotational proximity threshold is lower bounded by the cosine of the maximum angle between two adjacent faces of a polyhedral object, $\cos(\theta_{max})$, as set by the user in the virtual environment simulation:

$$\epsilon_{r_{LM}} = \max(\Delta t_{avg} \omega_{max}, \cos(\theta_{max})). \tag{6.14}$$



(a) Polygonal approximation of a round virtual object.

(b) Polygonal virtual object.

Figure 6.13: Typical commercial simulation packages use polygonal representations (in a planar virtual world) for both rounded objects and polygonal objects.

The local geometry described in this section approximates the virtual environment geometry in close proximity to the virtual tool. The approximation is derived such that it eliminates the collision detection bottleneck from the local model of interaction and improves the user's perception of tight clearances. Furthermore, it is able to interface the haptic device to a virtual environment regardless of the algorithms and data structures employed to generate the rigid multibody virtual world.

6.3 Local dynamics

Besides local geometry, local dynamics affect the realism of the haptic feedback provided to users. They allow the local model to account for the inertia of the virtual tool during the computation of the virtual interactions. Furthermore, the local dynamics allow the local model to compute physically-motivated impulsive forces that are guaranteed not to increase the kinetic energy of the proxy during collisions. In essence, the local dynamics provide an efficient implementation of the simulation and the control techniques developed in Chapters 4 and 5. Their presentation in this section emphasizes the approximations that they involve.

Local dynamics comprise the dynamics of a proxy of the virtual tool constrained by the ϵ-active local geometry presented in Section 6.2. As described in the following section, the ϵ-active local geometry is enforced through the impulse-augmented penalty approach introduced in Chapter 4. In the local model, this approach is based on a model of proxy contact derived from the rigid body contact model discussed in Section 4.2. The proxy contact model has three states: free motion, colliding contact, and resting contact. The proxy is said to be in *free motion* if it has no contacts (i.e., the separation distances between the proxy and all local constraints are strictly positive). The proxy is said to be in *colliding contact* if it has at least one new contact with negative normal contact velocity. Finally, the proxy is said to be in *resting contact* if it is neither in free motion nor in colliding contact. Using this proxy contact model, the constraints imposed on the virtual tool by the virtual environment are applied to users as detailed in the following section.

6.3.1 Local contact interactions applied to users

Haptic manipulation of linkages is incorporated by defining the virtual tool to be the entire articulated structure when the user holds one of its links[3] (see Figure 6.2) and by computing the proxy dynamics in configuration space. If the virtual tool is a rigid object, then its Cartesian space dynamics are its configuration space dynamics.

The computational load of the local model of interaction is much reduced compared to that of the virtual environment simulation. This is because local geometry makes local collision detection trivial and because only the virtual tool dynamics are simulated locally. During manipulation of linkages, numerical efficiency is further increased by approximating the configuration space linkage inertia $\mathbf{D}_{vt}$, the configuration space gravitational terms $\mathbf{G}_{vt}$, and the topological constraints with their values at the moment of the update. Furthermore, the Coriolis and centripetal effects are neglected, similar to work in [126, 128]. Thus, the proxy dynamics during resting contact are simulated by:

$$\widehat{\mathbf{D}}_{vt}\ddot{\mathbf{q}} + \widehat{\mathbf{G}}_{vt} = \sum_{i=1}^{c} \mathbf{J}_i^T \mathbf{F}_i + \mathbf{J}_h^T \mathbf{F}_h, \tag{6.15}$$

while its dynamics during colliding contact become:

$$\widehat{\mathbf{D}}_{vt}\dot{\mathbf{q}} = \widehat{\mathbf{D}}_{vt}\dot{\mathbf{q}}_0 + \mathcal{J}_c\mathbf{p}. \tag{6.16}$$

[3] By this definition, nearby contacts of all links must be sent to the local model.

In Equations (6.15) and (6.16), $\widehat{\mathbf{D}}_{vt}$ and $\widehat{\mathbf{G}}_{vt}$ are the values of $\mathbf{D}_{vt}$ and $\mathbf{G}_{vt}$, respectively, computed by the simulation at the moment of the update. Furthermore, c is the number of proxy contacts, $\mathbf{F}_i$ are penalty and friction forces, $\mathbf{p}$ is the collision impulse (computed as discussed in Section 4.4), $\mathbf{J}_i$ is the proxy Jacobian at the i-th proxy contact, $\mathbf{J}_h$ is the proxy Jacobian computed at the user's hand, $\mathbf{F}_h$ is the user-applied wrench, $\mathcal{J}_c = \left(\mathbf{J}_1^T \mathbf{n}_1 \ldots \mathbf{J}_i^T \mathbf{n}_i \ldots \mathbf{J}_c^T \mathbf{n}_c\right)^T$, and $\mathbf{n}_i$ is the local constraint normal at the i-th proxy contact. Passivity of proxy dynamics during collisions is guaranteed by assuming static constraints when resolving local collisions. This assumption prevents the simulation from inputting energy into the local model.

Finally, the proxy contact interactions are applied to users using the approximate SVD of the hand Jacobian $\mathbf{J}_h$, computed by the virtual environment simulation at the moment of the update:

$$\widehat{\mathbf{J}}_h = \widehat{\mathbf{U}} \widehat{\mathbf{\Sigma}}_1 \widehat{\mathbf{\Sigma}}_2 \widehat{\mathbf{V}}. \tag{6.17}$$

In Equation (6.17), $\widehat{\mathbf{\Sigma}}_1 \in \mathcal{R}^{6 \times r}$ is a matrix that has the singular values of $\widehat{\mathbf{J}}_h$ on its main diagonal and zeros elsewhere, $\widehat{\mathbf{\Sigma}}_2 \in \mathcal{R}^{r \times d}$ is a matrix whose elements are equal to one on the main diagonal and equal to zero elsewhere, and $r = rank(\widehat{\mathbf{J}}_h)$. Hence, the local coordinate invariant inverse of the linkage Jacobian at the user's hand is:

$$\widehat{\mathbf{J}}_h^{\#} = \widehat{\mathbf{D}}_{vt}^{-1} \widehat{\mathbf{V}} \widehat{\mathbf{\Sigma}}_2^T \left(\widehat{\mathbf{\Sigma}}_2 \widehat{\mathbf{V}}^T \widehat{\mathbf{D}}_{vt}^{-1} \widehat{\mathbf{V}} \widehat{\mathbf{\Sigma}}_2^T\right)^{-1} \left(\widehat{\mathbf{\Sigma}}_1^T \widehat{\mathbf{U}}^T \mathbf{M}_{hl} \widehat{\mathbf{U}} \widehat{\mathbf{\Sigma}}_1\right)^{-1} \widehat{\mathbf{\Sigma}}_1^T \widehat{\mathbf{U}}^T \mathbf{M}_{hl}, \tag{6.18}$$

where $\mathbf{M}_{hl}$ is the mass matrix of the link held by the user. Users feel an environment force:

$$\mathbf{F}_{env} = \widehat{\mathbf{J}}_h^{\#} \frac{\mathcal{J}_c^T \mathbf{p}}{\Delta t} \tag{6.19}$$

upon proxy contact, and they feel:

$$\mathbf{F}_{env} = \widehat{\mathbf{J}}_h^{\#} \sum_{i=1}^{c_a} \mathbf{J}_i^T \mathbf{F}_i \tag{6.20}$$

during proxy contact.

6.3.2 Local control

As discussed in Chapter 5, linkage inertia and the motion constraints due to linkage topology are imposed on users through device control. When users operate virtual linkages through the local model of interaction, the configuration space inertia of the linkage $\mathbf{D}_{vt}$ and the singular directions of $\mathbf{J}_h^T$ are approximated through their values in the virtual environment at the moment of the update. It follows that the inverse of the operational space inertia of the virtual tool at the user's hand $\mathbf{\Lambda}_h^{-1}$ and the directions of topological constraint $\mathbf{n}_{tc,i}$ are also approximated in the local model of interaction. Specifically, the local inverse of the operational space inertia of the virtual tool at the user's hand $\widehat{\mathbf{\Lambda}}_h^{-1}$ is given by:

$$\widehat{\mathbf{\Lambda}}_h^{-1} \quad = \quad \mathbf{J}_h \widehat{\mathbf{D}}_{vt}^{-1} \mathbf{J}_h, \tag{6.21}$$

and the control signal imposing the approximate topology on users is:

$$\mathbf{u}_{add} = \mathbf{M}_d \sum_{i=1}^{n_{tc}} \left(k_{tc} \widehat{\mathbf{n}}_{tc,i}^T \left(\mathbf{x}_{h_{vt}} - \mathbf{x}_h \right) + b_{tc} \widehat{\mathbf{n}}_{tc,i}^T \left(\dot{\mathbf{x}}_{h_{vt}} - \dot{\mathbf{x}}_h \right) \right) \widehat{\mathbf{n}}_{tc,i}, \tag{6.22}$$

where n_{tc} and $\widehat{\mathbf{n}}_{tc,i}$ are the number of topological constraints and the i-th topological constraint direction, respectively, computed by the virtual environment simulation at the moment of the update. Furthermore, recall from Chapter 5 that $\mathbf{M}_d$ is the mass matrix of the haptic device, k_{tc} and b_{tc} are the stiffness and damping of the topological constraints, $\mathbf{x}_h$ and $\dot{\mathbf{x}}_h$ are the body position and body velocity of the user's hand, and $\mathbf{x}_{h_{vt}}$ and $\dot{\mathbf{x}}_{h_{vt}}$ are the simulated body position and simulated body velocity of the virtual tool at the user's hand in the local model of interaction.

Equations (6.21)-(6.15) provide significant computational savings in the local model of the interaction and allow users to manipulate linkages with larger number of links. Realistic operation of virtual linkages using these approximations is demonstrated experimentally in the following section.

6.4 Experiments

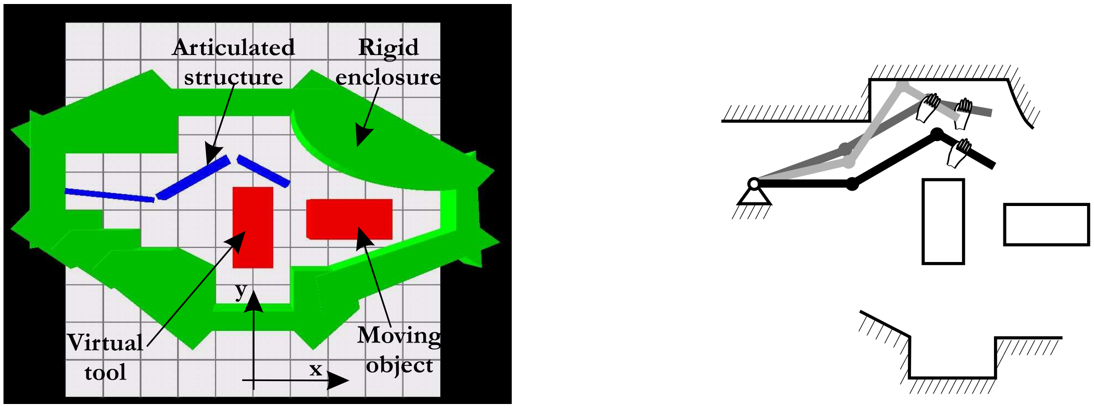
In this section, the performance of the proposed local model of interaction is compared to the performance of the intermediate representation proposed in [21,22]. The intermediate representation comprises the geometry of the contacts of the virtual tool at the moment of the update. Furthermore, stable interaction within a virtual environment connected to the haptic interface through the intermediate representation is ensured through smoothing forces at updates using constraint interpolation, as proposed in [100].

Five experiments are performed in this section, four of which are controlled experiments. All experiments are implemented on the planar haptic simulation system described in Chapter 3 using asynchronous communication. The first experiment validates the passivity of the proxy dynamics during collisions in the impulse-augmented penalty local model. The second experiment demonstrates that the ϵ-active geometry decreases the drift caused by local proxy deformation between the user-perceived constraints and the constraints in the virtual environment. The third experiment shows that the ϵ-active geometry improves the haptic perception in locally cluttered virtual environments. The fourth experiment establishes that higher contact stiffnesses are achievable in the local model when a four channel haptic controller [136] renders the virtual interactions to the user then when the impedance haptic device is directly coupled to the admittance local model, as done in [21]. The last experiment suggests that further improvements are required to increase the realism of the contact with dynamic objects.

6.4.1 Passive collision dynamics in the impulse-augmented penalty local model

The passivity of the proxy dynamics during collisions in the impulse-augmented penalty local model is investigated through user manipulations within the virtual environment shown in Figure 6.14(a) in its initial configuration. In the experiments described in this section, the user manipulates the distal link of the virtual linkage, as schematically depicted in Figure 6.14(b). The contact stiffness and damping are $K_{contact} = 15000$N/m and $B_{contact} = 100$N/(m/s), respectively. The contacts are perfectly plastic ($e = 0$) during a first example interaction, and perfectly elastic ($e = 1$) during a second example interaction. The number of simultaneous collisions, the kinetic energy of the proxy, and the environment wrenches are monitored during interaction.

The number of simultaneous collisions and the kinetic energy are plotted for both coefficients of restitution in Figure 6.15. Note that the kinetic energy of the proxy decreases during perfectly plastic collisions with the virtual environment (Figure 6.15(a)) and remains constant during perfectly elastic collisions with the environment (Figure 6.15(b)). These experimental results validate the passivity of the proposed collision resolution approach in the local model of interaction regardless of the contact restitution properties and of the number of simultaneous collisions. The passive collision resolution enhances the passivity of a simulation that computes impulsive and penalty interactions compared to a simulation that computes only penalty interactions. Therefore, the coupled stability of the haptic manipulation of rigid virtual tools increases within an impulse-augmented virtual environment compared compared to a penalties-only virtual world.

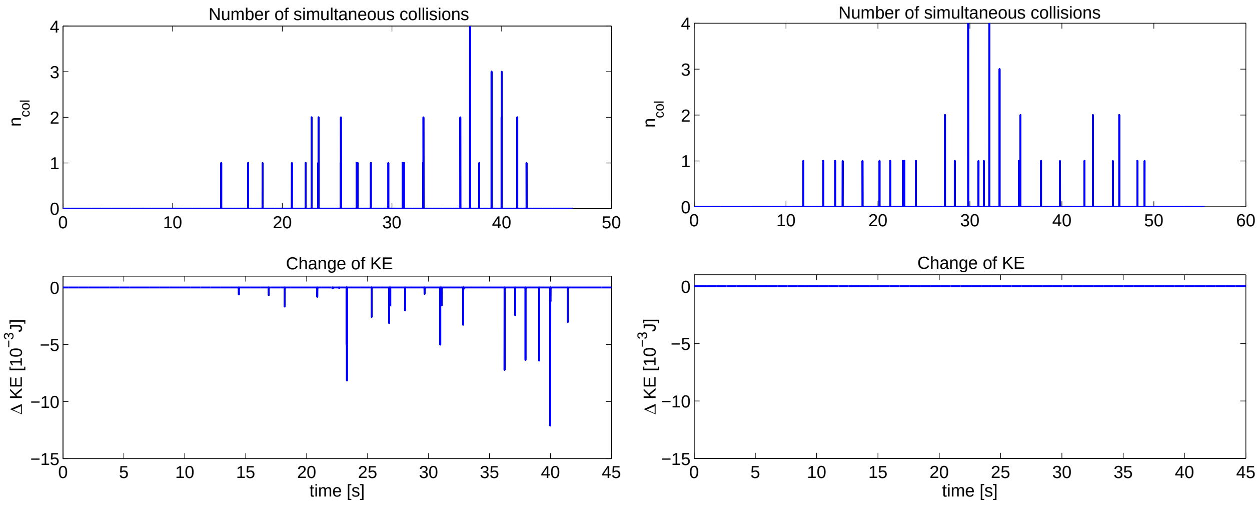


(a) Testbed virtual environment.

(b) Schematic of the interaction. Initial position is shown in black. Sample later positions are shown in shades of grey (the lightest shade indicates the latest configuration).

Figure 6.14: Experimental interaction used to investigate the passivity of the collision resolution approach proposed in this thesis.

The environment wrenches during the two experimental interactions are shown in Figure 6.16 for perfectly plastic collisions, and in Figure 6.17 for perfectly elastic collisions. Because the wrenches applied to users during collisions are much larger than those applied to users during contact (by up to four orders of magnitude), these figures display the environment wrenches using a logarithmic scale. Recall that the perceived rigidity of a virtual environment can be enhanced by allowing users to feel large forces upon contact [90, 125]. Hence, in addition to validating the passivity of the collision resolution, the experiments presented in this section confirm that the impulsive forces improve users' perception of rigidity within virtual worlds interfaced to the haptic



(a) Perfectly plastic contacts, $e = 0$.

(b) Perfectly elastic contacts, $e = 1$.

Figure 6.15: Passivity of proxy dynamics in an impulse-augmented penalty local model of interaction. The user operates the virtual linkage in Figure 3.3(b) from its distal link.

device through the proposed local model of rigid body interaction.

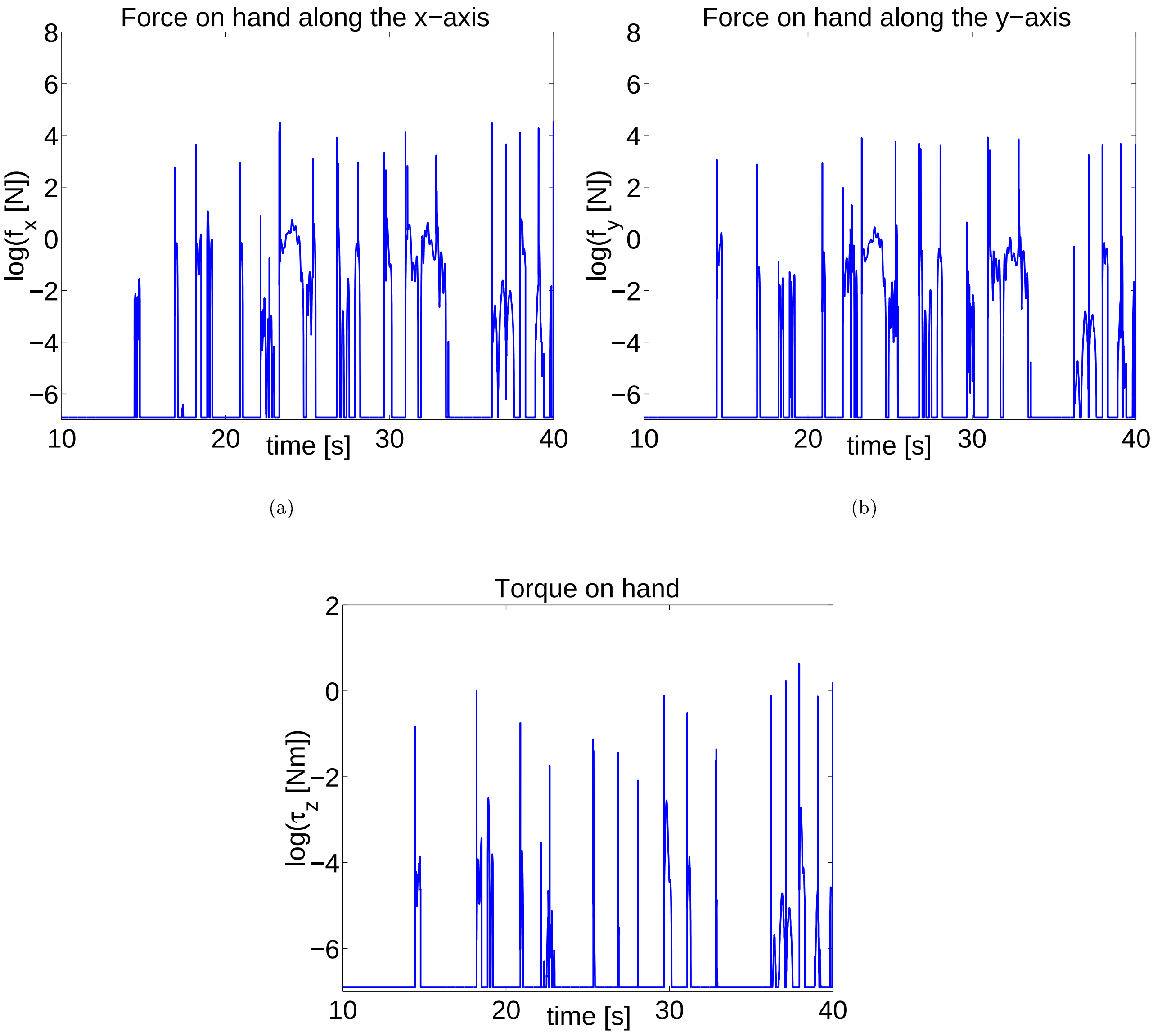
6.4.2 Transparency in the local model of interaction

The first experiment described in this section investigates the effect of local ϵ-active geometry on the transparency of haptic rendering of static rigid contacts. In this experiment, the user's action is represented by a constant wrench $\mathbf{F}_h = (0\text{N}\ 0.5\text{N}\ 0\text{Nm})^T$[4] pushing the rectangular virtual tool shown in Figure 6.18(a) towards the bottom-most horizontal wall of the rigid enclosure, as schematically depicted in Figure 6.18(b). The stiffness and the damping of the local contacts are $K_{contact} = 4000\text{N/m}$ and $B_{contact} = 30\text{N/(m/s)}$, respectively. These values represent the maximum contact impedance for which the interaction is stable in the intermediate representation (i.e., the virtual tool can be inserted in the tight-fitting hole at the bottom of the virtual world and remains stable upon being left there). Furthermore, collisions are considered perfectly plastic, i.e., $e = 0$. The device trajectories are plotted in Figure 6.19. These trajectories are obtained by interfacing the haptic device to the virtual world through: (i) the intermediate representation[5] ("IR"); (ii) the proposed local model including active geometry ("LM$_0$"); and (iii) the proposed local model including ϵ-active geometry within 5mm from the virtual tool ("LM$_5$").

The trajectories depicted in Figure 6.19 show the user's motion toward the virtual wall and the bounce once the virtual tool contacts the wall. Note that users perceive the virtual wall at different locations when the device

[4] The force is limited by the performance of the virtual environment (i.e., the virtual tool moves fast enough to pop through the wall for larger hand forces).

[5] Stable interaction has been ensured through smoothing forces at updates using constraint interpolation, as proposed in [100].

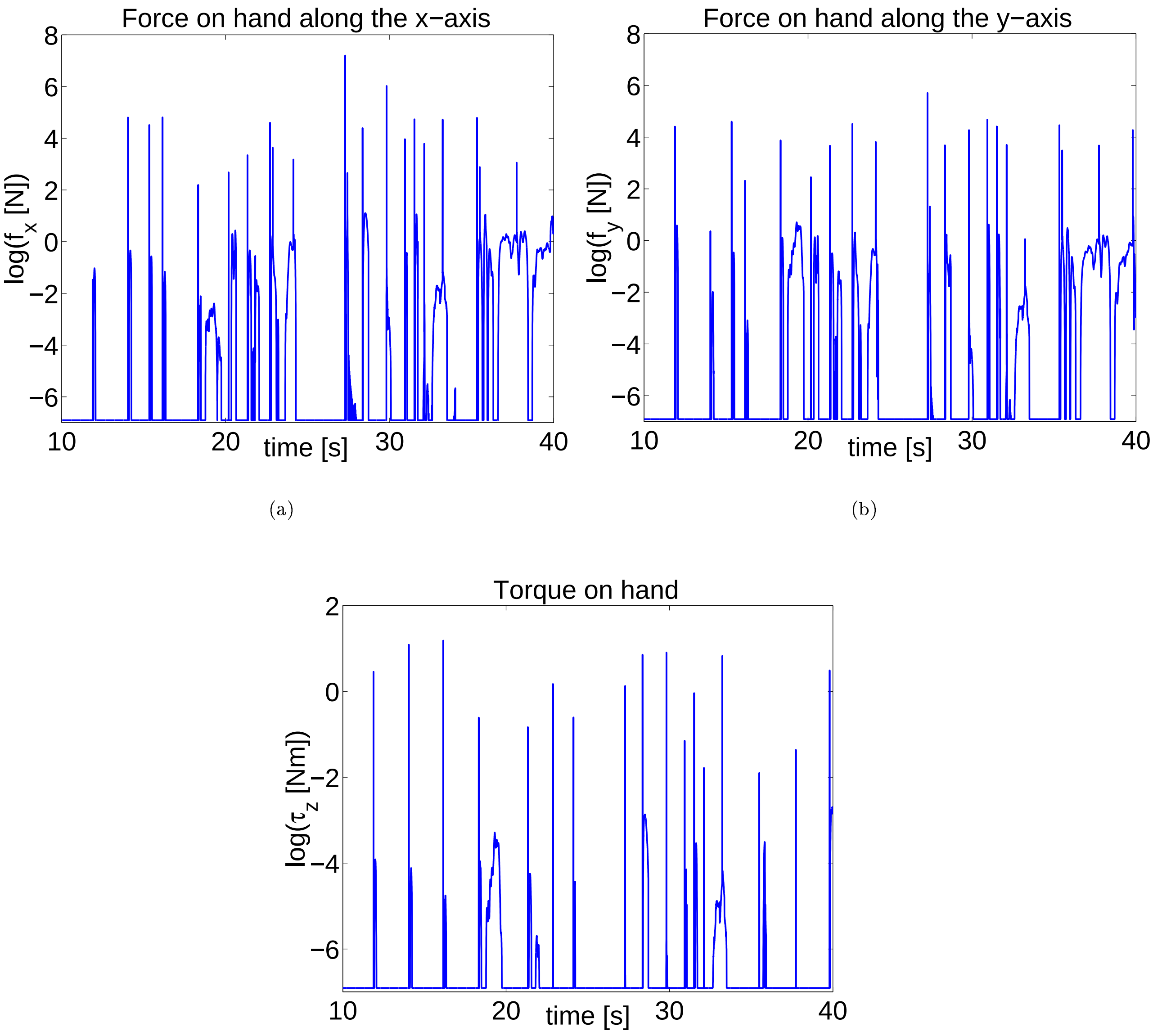


(a)

(b)

(c)

Figure 6.16: Environment wrenches applied to users during user manipulation of the distal link of the articulated object in Figure 6.14(a) in an impulse-augmented penalty local model of interaction with perfectly plastic contacts, $e = 0$.

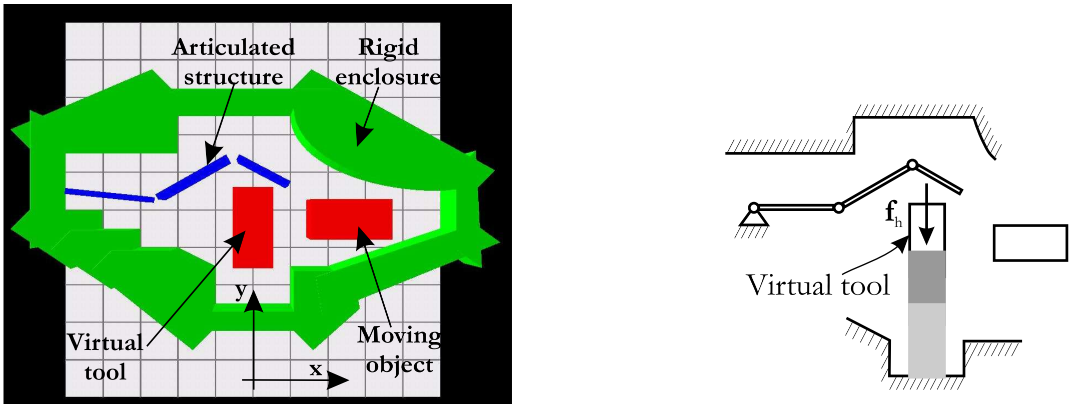


(a)

(b)

(c)

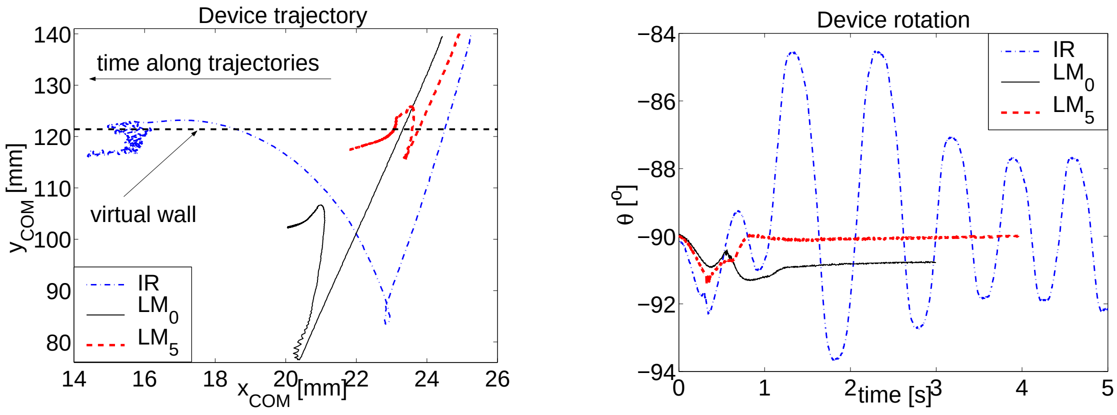
Figure 6.17: Environment wrenches applied to users during user manipulation of the distal link of the articulated object in Figure 6.14(a) in an impulse-augmented penalty local model of interaction with perfectly elastic contacts, $e = 1$.



(a) Testbed virtual environment.

(b) Schematic of the interaction. Initial position is shown in black. Sample later positions are shown in shades of grey (the lightest shade indicates the latest configuration).

Figure 6.18: Experiment used to investigate the haptic rendering of static contact.



(a)

(b)

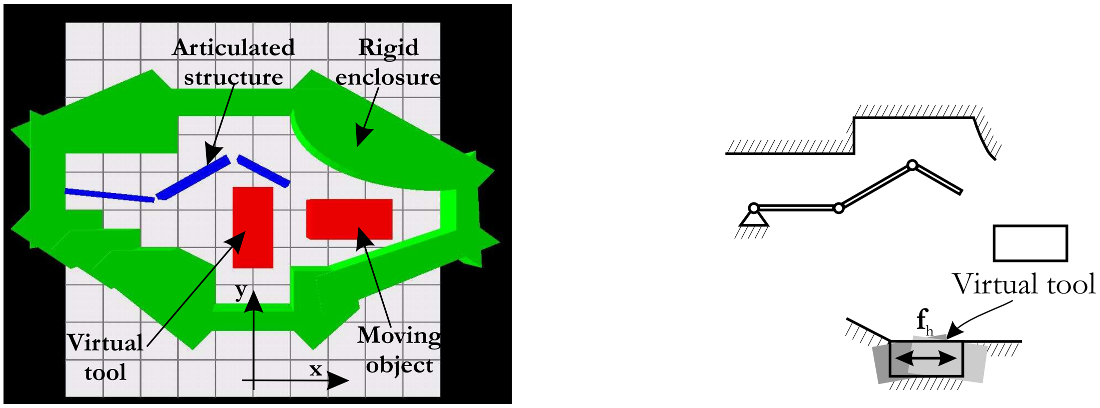
Figure 6.19: Device trajectories obtained when the virtual tool shown in Figure 6.18(a) is pushed with a constant force towards the bottom-most horizontal wall of the virtual environment, as depicted in Figure 6.18(b). The intermediate representation ("IR"), the local model including active geometry ("LM_0"), and the local model including ϵ-active geometry within 5mm from the virtual tool ("LM_5") interface the haptic device to the virtual environment simulation.

is interfaced to the virtual world through the intermediate representation, the local model with active geometry, and the local model with ϵ-active geometry. In particular, users perceive the virtual world at its location in the virtual environment, $y_{COM} = 116$mm, when ϵ-active geometry within 5mm of the virtual tool is included in the local model. Users perceive the virtual wall at $y_{COM} = 87$mm and at $y_{COM} = 80$mm in the intermediate representation and in the local model with active geometry, respectively[6]. The drift between the perceived location of the wall and its location in the virtual environment is caused by the smoothing techniques used to avoid force discontinuities at updates in the intermediate representation and in the local model, constraint interpolation and proxy deformation, respectively. Since users perceive no drift when interacting with the local model with ϵ-active geometry, the experiment illustrates that no proxy deformation occurs in this case. In turn, this means that constraints are sent to the local model before they become active. Hence, the experiment demonstrates that ϵ-active geometry diminishes the drift between the user-perceived position of the constraint and its position in the virtual environment. While the drift is dependent on the user-applied forces and the virtual world geometry, the experiment illustrates that it can be significantly reduced (by approximately 3.5cm in this case) by including in the local model ϵ-active constraints within a relatively small neighborhood of the virtual tool (5mm in this experiment). By alleviating this drift, the ϵ-active geometry increases the transparency of the interaction.

The second experiment demonstrates the role of the ϵ-active geometry in improving the perception of tight virtual clearances. In this experiment, the controlled interaction represents a peg-in-hole manipulation. The user rotates the rectangular virtual tool shown in Figure 6.20(a) by 90^o and inserts it into the hole at the bottom of the rigid enclosure that exactly fits the peg. The user then releases the peg and their hand is replaced by the wrench $\mathbf{F}_h = (1sin(5\pi)\text{N } 0\text{N } 0\text{N/m})^T$, i.e., the peg is shaken horizontally by a sinusoidally varying force, as schematically represented in Figure 6.20(b). As before, the stiffness and damping of the local contacts are $K_{contact} = 4000$N/m and $B_{contact} = 30$N/(m/s), respectively. The experimental device trajectories are shown in Figure 6.21. They are obtained by interfacing the haptic device to the virtual environment through: (i) the intermediate representation ("IR"); (ii) the present local model including active geometry ("LM$_0$"); and (iii) the local model including ϵ-active geometry within 5mm of the virtual tool ("LM$_5$").

The experimental trajectories illustrate that, depending on the technique employed to haptically render the virtual hole, the device travels different distances along the x direction at different locations along the y direction in the virtual world. The differences in the three trajectories result from two combined factors. First, the constraints (representing both the lateral and the top virtual walls) arrive with delay in the intermediate representation and in the local model with active geometry. Hence, not all constraints representing the hole geometry are avaialable in these models throughout the interaction and these models render only approximations of the hole geometry to the user. The constraints arrive before becoming active in the local model with ϵ-active geometry. Therefore, all constraints representing the hole geometry are available in this model throughout the interaction and this model renders the exact hole geometry to the user. Approximate

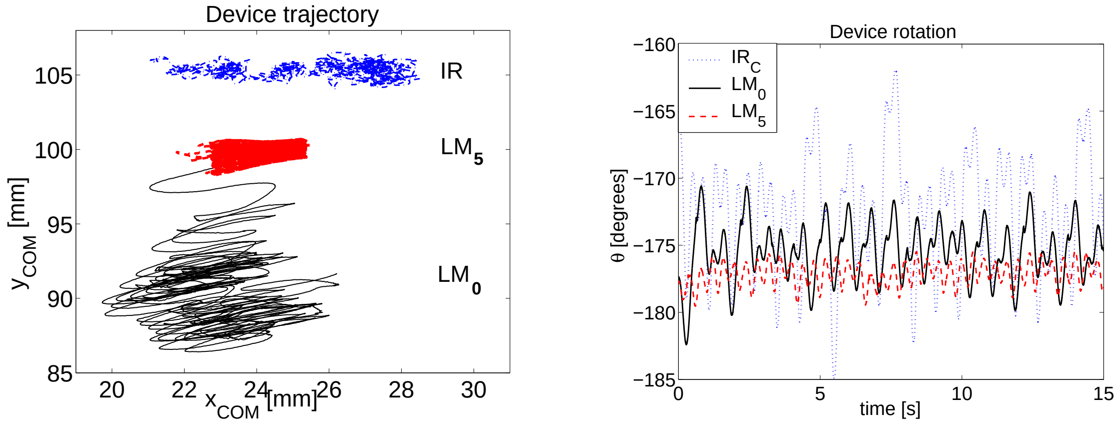
[6] The different locations are due to the slightly different starting positions of the device. These differences are caused by friction in the device and the limited stiffness of the controller used to drive the device to the initial position.



(a) Testbed virtual environment.

(b) Schematic of the interaction. Initial position is shown in black. Sample later positions are shown in shades of grey (the lightest shade indicates the latest configuration).

Figure 6.20: Experiment used to demonstrate the role of ϵ-active geometry in improving user's perception of tight virtual clearances.



(a)

(b)

Figure 6.21: Device trajectories obtained when the virtual tool shown in Figure 6.20(a) is rotated by 90^o, inserted into the tight-fitting hole at the bottom of the virtual world, and shaken horizontally with a sinusoidally varying force, as depicted in Figure 6.20(b). The intermediate representation ("IR"), the local model including active geometry ("LM_0"), and the local model including ϵ-active geometry within 5mm from the virtual tool ("LM_5") interface the haptic device to the virtual environment simulation.

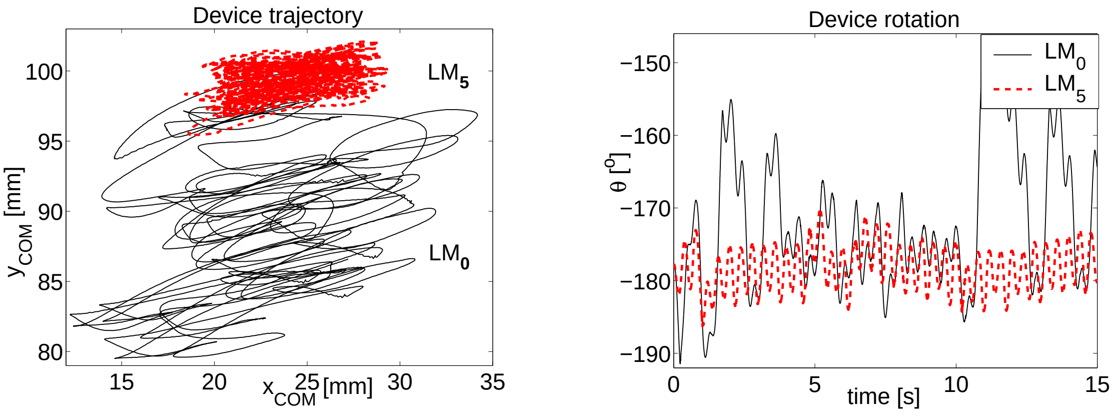
(i.e., incomplete) hole geometry accounts for the larger device rotation and for the larger free space haptically perceived by users (represented through larger device motion along the x direction) in the IR and the LM_0 trajectories in Figure 6.21. Second, the constraints are imposed on the device using only penalty forces in the intermediate representation and using impulsive and penalty forces in the local model of interaction. Approximate (incomplete) hole geometry together with the contact model explain the variances along the y axis in the device trajectories. In particular, when the virtual constraints are imposed through penalty forces (the IR trajectory), the device bounces and loses contact with the constraints. During free motion, the interface drift along the y direction is unbalanced and the device moves away from the horizontal wall of the hole, as illustrated in Figure 6.21(a). When the virtual constraints are imposed through impulsive and penalty forces (the LM_0 trajectory), the device maintains contact with the lateral constraints longer and friction in the lateral walls balances the interface drift. The interface motion towards the bottom of the hole is caused by the intermittent loss of contact with the bottom wall, which results in the horizontal constraints arriving late to the local model with active geometry. In turn, the delayed horizontal constraints cause local proxy deformation along the y-direction and allow the interface to drift towards the bottom of the hole.

6.4.3 The effect of the teleoperation controller on the achievable contact stiffness

To demonstrate the effect of the teleoperation controller on the achievable contact stiffness, the controlled peg-in-hole experiment depicted in Figure 6.20 is repeated using only the local model of the interaction and increasing the stiffness and the damping of the local contacts to $K_{contact} = 15000\mathrm{N/m}$ and $B_{contact} = 100\mathrm{N/(m/s)}$, respectively[7]. This stiffness is almost four times larger than the contact stiffness used in the previous experiments. Moreover, the controlled wrench applied on the device is $\mathbf{F} = (3sin(5\pi)\mathrm{N}\ 0\mathrm{N}\ 0\mathrm{N/m})^T$. The device trajectories obtained using both active ("LM_0") and predicted ("LM_5") geometry in the local model of the interaction are plotted in Figure 6.22.

The experimental trajectories show that much higher contact stiffness can be rendered to users when a four channel controller applies the virtual interactions to the user's hand then when the impedance haptic device is directly coupled to the admittance local model of interaction. Furthermore, these trajectories validate the positive effect of predicted geometry on the perception of tight clearances. Note that the device travels approximately 2cm more along both the x and the y axes and rotates more around the horizontal orientation of the virtual peg when the local model incorporates active geometry than when it incorporates predicted geometry within 5mm of the virtual tool. However, due to overshoot, the device penetrates into the constraints more than 5mm and the neighborhood selected for geometry prediction is not sufficient to eliminate the impact of the virtual environment delay on the interaction represented by the LM_5 trajectory. This impact is represented in the larger device motion along the x axis than permitted by the contact stiffness.

[7] Only the local model of interaction is used to interface the device to the simulation because the larger contact stiffness renders the interaction unstable if the device is directly coupled to the intermediate representation proposed in [21]



(a)

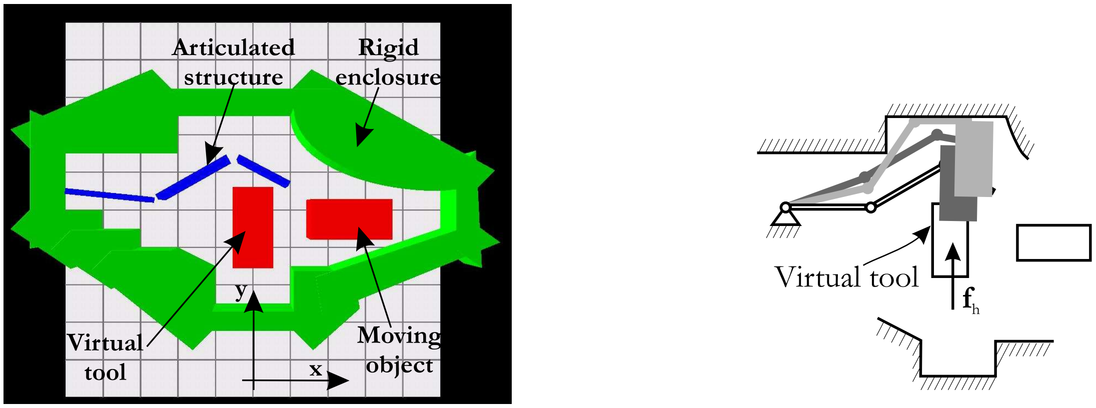
(b)

Figure 6.22: Device trajectories obtained when the virtual tool is inserted in the tight-fitting hole at the bottom of the virtual world shown in Figure 6.20(a) and shaken horizontally with a sinusoidally varying force, as depicted in Figure 6.20(b). The local model including active geometry ("LM_0") and the local model including predicted geometry within 5mm from the virtual tool ("LM_5") interface the haptic device to the virtual environment simulation.

6.4.4 Dynamic contact

The controlled manipulation described in this section investigates the realism of the haptic interaction with moving virtual objects using the virtual environment depicted in Figure 6.23(a). To allow comparison between the local model and the intermediate representation, the virtual contacts have stiffness $K_{contact}$ = 4000N/m and damping $B_{contact}$ = 30N/(m/s). Moreover, the user's hand is replaced by a constant wrench $\mathbf{F}_h = (0\text{N}\ 0.1\text{N}\ 0\text{N/m})^T$ that pushes the virtual tool until the last link of the virtual linkage hits the top-most horizontal wall of the enclosure (placed at y_{COM} = 181mm), as schematically depicted in Figure 6.23(b). Figure 6.24 depicts the device trajectories obtained by interfacing the haptic device to the virtual world through: (i) the intermediate representation ("IR"); (ii) the local model including predicted geometry within 5mm of the virtual tool and static constraints ("LM_{SC}"); and (iii) the local model including predicted geometry within 5mm of the virtual tool and moving (kinematic) constraints ("LM_{KC}"). In the local model, the kinematic constraints move along the contact normal with the velocity that they have in the virtual environment at the moment of the update.

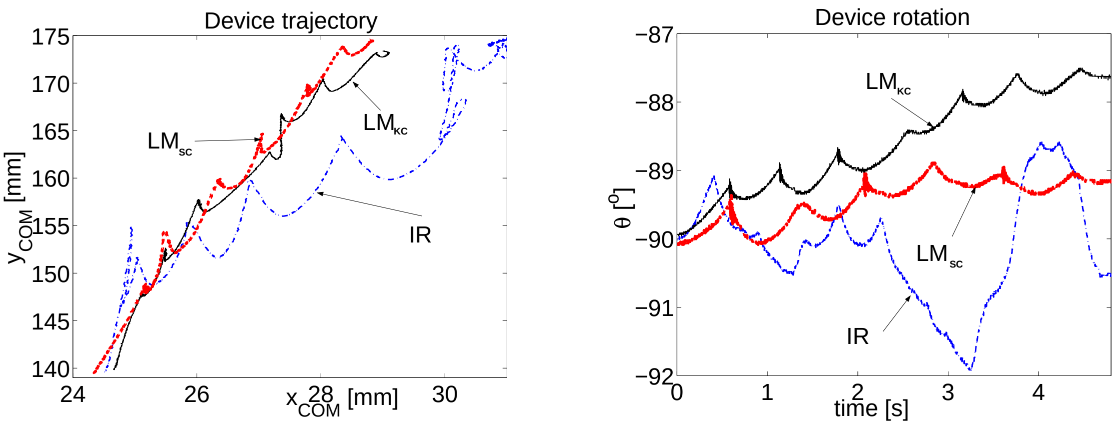
Several remarks result from the experimental trajectories. First, the haptic device (i.e., the user's hand) loses contact with the virtual link several times during pushing, regardless of the technique used to interface the device to the simulation. This non-physical behavior is demonstrated by the bounces recorded in all trajectories in Figure 6.24. Second, the device bounces more when connected to the virtual world through the intermediate representation then when connected through the local model of interaction. Third, the bounces are similar when the local model of interaction interfaces the device to the virtual environment, regardless of whether



(a) Testbed virtual environment.

(b) Schematic of the interaction. Initial position is shown in black. Sample later positions are shown in shades of grey (the lightest shade indicates the latest configuration).

Figure 6.23: Experiment used to investigate the realism of the haptic rendering of dynamic contact.



(a)

(b)

Figure 6.24: Device trajectories obtained when the virtual tool pushes the last link of the virtual linkage in Figure 6.23(a) with a constant force, as shown in Figure 6.23(b). The intermediate representation ("IR"), the local model including predicted geometry and static constraint ("LM_{SC}"), and the local model including prdicted geometry and moving (kinematic) constraints ("LM_{KC}") interface the haptic device to the virtual environment simulation.

the local constraints are static or kinematic. Two conclusions can be drawn from these experimental results. First, further improvements are required both in the intermediate representation and in the local model of interaction to increase the realism of dynamic contact, i.e., to maintain contact while interacting with moving virtual objects. Second, the local contact model enhances the stability of dynamic contact compared to the intermediate representation. This improvement is due to the local modeling of collisions and is not influenced by the use of kinematic local constraints.

This chapter has proposed the first local model of rigid body interaction that has been used to constrain both the translation and the rotation of a haptic device. This model comprises geometry and dynamics techniques that enable efficient and transparent haptic rendering of rigid body motion with constraints. Local geometry maintains force continuity at model updates through implementing novel techniques like local proxy deformation and ϵ-active geometry. Local dynamics efficiently implements the simulation and control methods developed in Chapters 4 and 5. The model has been validated by comparing its performance to the performance of existing approaches through experiments performed using a planar haptic interaction system. The experiments confirm that the local model proposed in this thesis increases the stability and the transparency of rigid body interaction within multibody virtual environments. This model allows users to feel collisions upon contact, stiffer constraints during contact, and topological constraints during linkage manipulation. The model also allows haptic manipulation of virtual linkages from any user-selected link, as demonstrated by the experiments presented in Chapter 5. In addition, the proposed local model of interaction provides a general purpose methodology for interfacing a haptic device to a virtual environment simulation, as it imposes no restrictions on the data structures and the algorithms used to generate the rigid multibody virtual world.

Chapter 7

Conclusion

This thesis has proposed a novel approach to applying realistic forces to users while they operate rigid virtual tools within multibody virtual worlds. In this approach, an original model of unilateral contact significantly increases the stability and the perceived rigidity of virtual contacts. A new model of bilateral contact and a coordinate invariant mapping of interactions from configuration space to operational space enable users to freely manipulate virtual linkages in addition to virtual objects. Fast local approximations of rigid body interaction interface a haptic device to a virtual environment simulation in a manner transparent to the application developer. As a result, convincing force feedback can easily be added to virtual prototyping and design applications, in a modular fashion and without a detailed consideration of the haptic simulation and control algorithms. The specific contributions to haptic rendering of rigid body motion with constraints of the thesis are summarized in the following section.

7.1 Summary of Contributions

- **Rigid contact modeling**

  A novel model of unilateral rigid body contact has been introduced that enables users to feel collisions. In the proposed model, contacts are infinitely stiff when they arise and have limited stiffness thereafter. To achieve this effect, the model combines constraint-based collision resolution with penalty-based contact resolution. This allows interaction forces and impulses to be computed and applied to users at the high fixed rates imposed by the haptic controller. The approach eliminates the need for constraint stabilization following collision resolution.

  A new collision resolution method has been developed that allows multiple collisions to be resolved simultaneously such that the kinetic energy of the multibody system never increases during collisions. The method is shown to apply equally to independent and to overdetermined constraints, as it implicitly eliminates constraint overdeterminancy.

  A physically motivated approach has been proposed to incorporate the limitations of the haptic device when impulses are applied to the user. The approach is consistent with conservation of energy principles and is made possible by the fact that constraints need not be stabilized after impact. Furthermore, the approach allows graceful degradation of performance when device capabilities diminish by maximizing the user's hand kinetic energy dissipated during impacts subjected to device capabilities.

A friction model developed in computational mechanics [61] has been used for the first time for haptic rendering of dry friction. Simulations and experiments have shown that this model renders a richer set of sticking contact characteristics than prior methods for haptic rendering of Coulomb friction. These characteristics are easily adjusted through a parameter with physical significance.

- **Haptic manipulation of linkages**

  Realistic manipulation of virtual linkages from any user-selected link and through singularities has been enabled through the development of new simulation and control methods. To increase computational efficiency and to eliminate the need for constraint stabilization, the dynamics of the multibody system have been simulated in configuration space. To enable compelling and unrestricted operation of virtual linkages, the configuration space dynamics have been mapped to operational space and applied to users through control. A four channel teleoperation controller imposes the unilateral constraints on users, while an augmented impedance controller applies the linkage inertia and the bilateral (topological) constraints to users. The coordinate invariant mapping of the linkage dynamics from configuration space to operational space and the representation of the topological constraints at the user's hand through control contribute new developments to haptic manipulation of virtual linkages.

  The coordinate invariant representation of linkage contacts at the user's hand is based on the theory of weighted generalized inverses developed in robotics [46] for purposes of hybrid control. While the mapping itself is not a new theoretical development, it has not been used in prior simulation research. The topological constraints are imposed on users through penalizing their departure from the configuration manifold of the virtual linkage through device control. The penalties embody the directions along which the virtual linkage has infinite structural stiffness at the user's hand. Simulations and experiments have been used to demonstrate that the topological penalties successfully impose the bilateral constraints on users.

- **Fast local approximations of rigid body interaction**

  Local models of rigid body interaction have been considered in prior research [21]; however, this work introduces the first local model that has been used to constrain both the translation and the rotation of the haptic device[1]. Five new features distinguish the proposed local model of rigid body interaction from existing models. First, the proposed local model uses local proxy deformation to avoid destabilizing force discontinuities at model updates. Experiments performed in a planar testbed virtual environment

[1] Device rotation is typically more challenging to constrain because, in typical simulation packages, rigid body contact in concave areas of the virtual environment is represented through a large number of vertex-face contacts whose stiffnesses compound such that they result in larger increases in the rotational stiffness than in the translational stiffness of the virtual environment [85].

demonstrate that local proxy deformation successfully maintains interaction stability in contact configurations that present challenges for prior local models (like peg-in-hole manipulations [21]). Second, the model comprises ϵ-active geometry, which has been demonstrated to increase the realism of the haptic manipulation of the virtual tool through small clearances. Third, the proposed local model implements the impulse-augmented penalty model of rigid contact developed in this thesis. Hence, it computes impulsive forces upon contact and penalty and friction forces during contact. The impulsive forces increase the stability of the local contacts. Fourth, the proposed local model includes a dynamic proxy in addition to local geometry. The dynamic proxy allows a four channel haptic controller [136] to be used for transmitting wrenches between the device and the proxy and for coordinating positions between them. Compared to the virtual coupler [3, 30], the four channel controller applies the locally computed impulsive forces to the user's hand, thus enabling users to feel collisions and improving the perceived rigidity [90, 125] of the virtual contacts. Compared to directly coupling an impedance device to an admittance simulation [21], the four channel controller allows users to feel stiffer contacts. Lastly, the proposed local model of interaction imposes no restrictions on the data stuctures and the dynamic response algorithms used to generate the virtual environment. Therefore, the model can be used to add force feedback to interactive virtual worlds generated using any commercial simulation package.

7.2 Future Work

The results of this work can be extended and its limitations addressed in several ways:

- **Improved contact models**

  Compared to prior models used for haptic rendering of rigid contact, the model developed in this thesis constrains the user's hand trajectory closer to the trajectory imposed by a real rigid wall. However, the performance of this model is little improved during contact with dynamic virtual environments. This is expected, since the model is derived from a penalty representation of static contact. More realistic models of dynamic contact can be developed that include inertial properties in addition to stiffness and damping. Such impedance-type representations of rigid contact may be developed that require minimal additional computational effort and more accurately embody dynamic contact.

- **Collisions with friction and arbitrary coefficients of restitution**

  The present collision resolution approach assumes frictionless collisions, all with the same coefficient of restitution. This assumption has successfully increased the stability and the perceived rigidity of the virtual contacts. However, it cannot be used to render rolling, sticking, or reversed sliding that might occur after collision. The development of a collision resolution method compatible with the fixed haptic rate requirements would allow such phenomena to be rendered to users, thus increasing the realism of the haptic manipulation of rigid objects within multibody virtual worlds.

- **Local geometry prediction based on user's intent**

  Local models of interaction are necessary for allowing haptic collaboration over internet. This work has been the first to allow rigid body manipulation using such a model. However, only crude geometry has been included in the proposed local model. As environment geometry is significant during the interaction with rigid virtual objects, improved prediction of local geometry is likely to result in enhanced realism of the interaction. The enhancement will allow the local model to suitably render more complex virtual environments, i.e., virtual environments with larger computational delays and more complex contact geometries. One promising approach to enriching the local geometry is contact prediction based on user intent rather than based on spatial proximity. Various mechanisms can be used to predict user intent, from simple ones (like user velocity) to others more complex (like Hidden Markov Models).

- **Virtual geometry for haptics**

  A wider spread of haptic applications requires the availability of general purpose interfaces between haptic devices and rigid multibody simulations like the local model proposed in this thesis. Such interfaces allow application developers to add force feedback to virtual environments without requiring them to have detailed knowledge of the haptic simulation and control algorithms. However, the quality of the haptic feedback provided by such interfaces depends on the suitability of the data structures used in the virtual environment simulation. Present data structures result in perceptual artifacts during haptic simulations, since they are typically not able to maintain contact coherence during contacts that involve concave surfaces. In this work, such limitations have been alleviated through contact clustering and through maintaining temporal coherence based on spatial coherence. However, this technique breaks down if the user moves fast in highly concave regions. Geometric data structures that can be used to maintain rigid body contact coherence must be developed to eliminate such artifacts.

- **Alternative collision rendering techniques and user studies**

  In this work, collisions have been rendered to users through impulsive forces that dissipate the kinetic energy predicted by the coefficient of restitution over one simulation step. This approach is based on the assumption that impacts are instantaneous events. Nonetheless, alternative rendering approaches might suitably approximate the instantaneous nature of collisions while allowing users to perceive other contact characteristics in addition to rigidity. Sinusoidally varying impulsive forces might allow users to distinguish between wood on wood or wood on metal collisions, for example. Future user studies need to be performed to identify salient features of the impulsive forces that can be manipulated in order to allow users to perceive such differences.

  Furthermore, the model used in this work to render dry friction has been shown adequate for displaying various stiction behaviors. Whether these behaviors can be associated to meaningful contact characteristics must be investigated through future user studies.

- **User studies**

  The haptic rendering techniques developed in this work have been shown to enable more transparent manipulations of rigid virtual objects than prior methods. The transparency of the interaction has been assessed via comparing physical outcomes of virtual manipulations implemented through varying modeling approaches to physical outcomes of real manipulations. No attempt has been made to assess the performance of the proposed methods via perceptual studies. This is partly because the motivation for some of the new techniques (such as the use of impulsive forces for haptic rendering of rigid contact) has been provided by existing user studies [125], [90]. Nevertheless, human perception is key to evaluating any haptic rendering approach. Therefore, future user studies are necessary to further validate the advantages of the methods developed in this work.

- **Guaranteed stable interaction**

  Lastly, the present work does not guarantee stable interaction within arbitrary rigid multibody virtual environments. This is because contacts are resolved using traditional penalty techniques. These techniques are conditionally stable and render environment stiffnesses that depend on contact geometry. Hence, their stability cannot be guaranteed for arbitrary virtual environments. This limitation needs to be addressed by developing a new concept of stiffness for rigid body constraints that breaks away from combining point stiffnesses into a resultant rigid body contact stiffness. The successful development of such a concept hinges on a physically meaningful metric for defining distances in configuration space.

At the end of this work, rigid body interaction within arbitrary virtual environments continues to be a challenging endeavor. The developments presented in this thesis have led to increased stability and enhanced realism of the force feedback applied to users while they manipulate virtual objects or virtual linkages. However, they do not guarantee that the interaction will be free of perceptual artifacts or stable regardless of the complexity of the virtual world. Further work is required to provide such guarantees and to transform haptic devices into widely used computer interfaces.

Bibliography

[1] M. Abadie. Dynamic Simulation of Rigid Bodies: Modeling of Frictional Contact. In B. Brogliato, editor, *Impacts in Mechanical Systems: Analysis and Modeling*, pages 62–144. Springer Verlag, Berlin, Lecture Notes in Physics, 2000.

[2] Y. Adachi, T. Kumano, and K. Ogino. Intermediate Representation for Stiff Virtual Objects. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pages 203–210, Research Triangle Park, NC, 1995.

[3] R.J. Adams and B. Hannaford. Stable Haptic Interaction with Virtual Environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, June 1999.

[4] S. Ahmed, H.M. Lankarani, and M.F.O.S. Pereira. Frictional Impact Analysis in Open-Loop Multibody Mechanical Systems. *ASME Journal of Mechanical Design*, 121:119–127, March 1999.

[5] M. Anitescu and G.D. Hart. A Hard-Constraint Time-Stepping Approch for Rigid Multibody Dynamics with Joints, Contact, and Friction. In *Proceedings of the 2003 Conference on Diversity in Computing*, pages 34–41, Atlanta, GA, 2003.

[6] M. Anitescu and G.D. Hart. A Constraint-Stabilized Time-Stepping Approach for Rigid Multibody Dynamics with Joints, Contact and Friction. *International Journal for Numerical Methods in Engineering*, in press 2004.

[7] M. Anitescu and F.A. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems. *ASME Nonlinear Dynamics*, 14:231–247, 1997.

[8] M. Anitescu, D.E. Stewart, and F.A. Potra. Time-Stepping for Three-Dimensional Rigid Body Dynamics with Friction. *Computer Methods in Applied Mechanics and Engineering*, 177(3-4):183–197, 1999.

[9] O. Astley and V Hayward. Multirate Haptic Simulation Achieved By Coupling Finite Element Meshes Through Norton Equivalents. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 989–994, Leuven, Belgium, 1998.

[10] D. Baraff. Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies. *Computer Graphics (Proceedings of the 1989 SIGGRAPH)*, 23:223–231, 1989.

[11] D. Baraff. Issues in Computing Contact Forces for Non-Penetrating Rigid Bodies. *Algorithmica*, 10(2-4):292–352, 1993.

[12] D. Baraff. Fast Contact Force Computation for Non-penetrating Rigid Body Simulation. *Computer Graphics (Proceedings of the 1994 SIGGRAPH)*, 29:23–34, 1994.

[13] D. Baraff. Interactive Simulation of Solid Rigid Bodies. *IEEE Computer Graphics and Applications*, 15:63–75, 1995.

[14] F. Barbagli, D. Prattichizzo, and J.K. Salisbury. Dynamic local models for stable multi-contact haptic interaction with deformable objects. In *Proceedings of 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 109–116, Los Angeles, CA, 2003.

[15] A.A. Barhorts and L.J. Everett. Contact/Impact in Hybrid Parameter Multiple Body Mechanical Systems. *ASME Journal of Dynamic Systems, Measurement, and Control*, 117:559–569, 1995.

[16] R. Barzel and A.H. Barr. A Modeling System Based on Dynamic Constraints. *Computer Graphics (Proceedings of the 1994 SIGGRAPH)*, 22:179–188, 1988.

[17] C. Basdogan, C.H. Ho, and M.A. Srinivasan. A Ray-Based Haptic Rendering Technique for Displaying Shape and Texture of 3D Objects in Virtual Environments. In *Proceedings of ASME Dynamics Systems and Control Division*, volume DSC-61, pages 77–84, 1997.

[18] J.A. Battle and S. Cardona. The Jamb (Self-Locking) Process in Three-Dimensional Collisions. *ASME Journal of Applied Mechanics*, 65:417–423, June 1998.

[19] J. Baumgarte. Stabilization of Constraints and Integrals of Motion in Dynamical Syatems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.

[20] A. Ben-Israel and T.N.E. Graville. *Generalized Inverses: Theory and Applications*. New York:Wiley, 1974.

[21] J.M Berkelman. *Tool-Based Haptic Interaction with Dynamic Physical Simulations using Lorentz Magnetic Levitation*. PhD thesis, Carnegie Mellon University, 1999.

[22] P.J. Berkelman, R.L. Hollis, and D. Baraff. Interaction with a Realtime Dynamic Environment Simulation using a Magnetic Levitation Haptic Interface Device. In *Proceedings of the 1999 International Conference on Robotics and Automation*, pages 3261–3266, Detroit, Michigan, 1999.

[23] V. Bhatt and J. Koechling. Partitioning the Parameter Space According to Different Behaviors During Three-Dimensional Impacts. *ASME Journal of Applied Mechanics*, 62(3):740–746, 1995.

[24] V. Bhatt and J. Koechling. Three-Dimensional Frictional Rigid-Body Impact. *ASME Journal of Applied Mechanics*, 62(4):893–898, 1995.

[25] C.S. Bonaventura and K.W. Jablokow. An O(N) Modular Algorithm for the Dynamic Simulation of Robots Constrained by a Single Contact. *The IEEE Transactions on Systems, Man, and Cybernetics: Part C*, 32(4):406–415, 2002.

[26] R.M. Brach. Rigid Body Collision. *ASME Journal of Applied Mechanics*, 56:133–138, 1989.

[27] B. Brogliato. *Nonsmooth Mechanics, Second Edition*. Springer-Verlag, London, 1999.

[28] J.M Brown. *Passive Implementation of Multibody Simulation for Haptic Display*. PhD thesis, Nothwestern University, 1995.

[29] J.M Brown. A Theoretical and Experimental Investigation into the Factors Affecting the Z-width of a Haptic Display. Master's thesis, Nothwestern University, 1998.

[30] J.M. Brown and J.E. Colgate. Physics-based Approach to Haptic Display. In *Proceedings of the Topical Workshop on Virtual Reality, International Symposium on Measurement and Control in Robotics*, volume 1, pages 101–106, Houston, TX, 1994.

[31] J.M. Brown and J.E. Colgate. Minimum Mass for Haptic Display Simulations. In *Proceedings of the 1998 ASME International Mechanical Engineering Congress and Exhibition*, pages 85–92, Nashville, TE, 1997.

[32] M. Buck and E. Schomer. Interactive Rigid Body Manipulation with Obstacle Contacts. *Journal of Visualization and Computer Animation*, 9(4):243–257, 1998.

[33] S. Cameron. Enhancing GJK: Computing Minimum and Penetration Distance between Convex Polyhedra. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3112–3117, 1997.

[34] B. Chang and J.E. Colgate. Real-Time Impulse-Based Simulation of Rigid Body Systems for Haptic Display. In *Proceedings of the 1997 ASME International Mechanical Engineering Congress and Exhibition*, pages 1–8, Dallas, Texas, 1997.

[35] A. Chatterjee and A. Ruina. A New Algebraic Rigid Body Collision Law based on Impulse Space Considerations. *ASME Journal of Applied Mechanics*, 65(4):939–951, 1998.

[36] J. Chen, C. DiMattia, R.M. Taylor II, M. Falvo, P. Thiansathaporn, and R. Superfine. Sticking to the Point: a Friction and Adhesion Model for Simulated Surfaces. *Proceedings of the ASME Dynamic Systems and Control Division*, DSC-61:167–171, 1997.

[37] M. Cline and D. Pai. Stabilization for Rigid Body Simulation with Contact and Constraints. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3744–3751, Taipei, Taiwan, September 2003.

[38] J.D. Cohen, M.C. Lin, D. Manocha, and M.K. Ponamgi. I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments. In *Symposium on Interactive 3D Graphics*, pages 189–196, Monterey, CA, August 1995.

[39] J.E. Colgate. Coupled Stability of Multiport Systems - Theory and Experiments. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, 116(3):419–428, 1994.

[40] J.E. Colgate and J.M. Brown. Factors Affecting the Z-width of a Haptic Display. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 3205–3210, San Diego, CA, 1994.

[41] J.E. Colgate and G.G. Schenkel. Passivity of a Class of Sampled-Data Systems: Application to Haptic Interfaces. *Journal of Robotic Systems*, 14(1):37–47, 1997.

[42] J.E. Colgate, M.C. Stanley, and J.M. Brown. Issues in the Haptic Display of Tool Use. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 140–145, Pittsburgh, PA, 1995.

[43] D. Constantinescu, I. Chau, L. Filipozzi, S.P. DiMaio, and S.E. Salcudean. Haptic Rendering of Planar Rigid-Body Motion using a Redundant Parallel Mechanism. In *Proc 2000 IEEE Int Conf Rob Autom*, pages 2440–2445, Berkeley, Ca, 2000.

[44] D. Constantinescu, S.E. Salcudean, and E.A. Croft. Haptic Feedback using Local Models of Interaction. In *11th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 416–421, Los Angeles, Ca, 2003.

[45] P.R. Dahl. A Solid Friction Model. Technical Report TOR-158(3107-18), The Aerospace Corporation, El Segundo, CA, May 1968.

[46] K.L. Doty, C. Melchiorri, and C. Bonivento. A Theory of Generalized Inverses applied to Robotics. *The International Journal of Robotics Research*, 12(1):1–19, February 1993.

[47] P.E. Dupont. The Effect of Coulomb Friction on the Exstence and Uniqueness of the Forward Dynamics Problems. In *The IEEE Proceedings of the 1992 International Conference on Robotics and Automation*, pages 1442–1447, Nice, France, 1992.

[48] R. Featherstone. *Robot Dynamics Algorithms*. Boston:Kluwer, 1987.

[49] R. Featherstone. A Divide-and-Conquer Articulated-Body Algorithm for Parallel $o(\log n)$ Calculation of Rigid-Body Dynamics: Part 1:Basic Algorithm. *The International Journal of Robotics Research*, 18(9):867–875, 1999.

[50] R. Featherstone. A Divide-and-Conquer Articulated-Body Algorithm for Parallel $o(\log n)$ Calculation of Rigid-Body Dynamics: Part 2: Trees, Loops, and Accuracy. *The International Journal of Robotics Research*, 18(9):876–892, 1999.

[51] J. Garcia de Jalon and E. Bayo. *Kinematics and Dynamics Simulation of Multibody Systems: The Real-Time Challenge*. Springer-Verlag, 1993.

[52] B. Gillespie. *Haptic Display of Systems with Changing Kinematic Constraints: The Virtual Piano Action*. PhD thesis, Stanford University, 1996.

[53] R.B. Gillespie. Kane's Equations for Haptic Display of Multibody Systems. *Haptics-e*, 3(2):1–20, 2003.

[54] R.B. Gillespie and M.R Cutkosky. Interactive Dynamics with Haptic Display. *Advances in Robotics, Mechatronics, and Haptic Interfaces, ASME*, DSC-49:65–72, 1993.

[55] B.G. Gilmore and D.A. Streit. A Rule Based Algorithm to Predict the Dynamic Behavior of Mechanical Part Orienting Operations. In *Proceedings of IEEE Conference on Robotics and Automation*, 1988.

[56] B.J. Gilmore and R.J. Cipra. Simulation of Planar Dynamic Mechanical Systems with Changing Topologies - Part 1: Characterization and Prediction of Kinematic Constraint Changes. *Journal of Mechanical Design, Transactions of the ASME*, 113:70–76, March 1991.

[57] S. Goyal, E.N. Pinson, and F.W. Sinden. Simulation of Dynamics of Interacting Rigid Bodies Including Friction I: General Problem and Contact Model. *Engineering with Computers*, 10:162–174, 1994.

[58] S. Goyal, E.N. Pinson, and F.W. Sinden. Simulation of Dynamics of Interacting Rigid Bodies Including Friction II: Software System Design and Implementation. *Engineering with Computers*, 10:175–195, 1994.

[59] A. Gregory, A. Mascarenhas, S. Ehmann, M. Lin, and D. Manocha. Six Degree-of-Freedom Haptic Display of Polygonal Models. In *IEEE Proceedings on Visualization*, pages 139–146, 2000.

[60] P.J. Hacksel and S.E. Salcudean. Estimation of Environment Forces and Rigid-Body Velocities using Observers. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, pages 931–936, San Diego, CA, May 1994.

[61] D.A.Jr. Haessig and B. Friedland. On the Modeling and Simulation of Friction. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, DSC-113:354–362, September 1991.

[62] J.K. Hahn. Realistic Animation of Rigid Bodies. *Computer Graphics (Proc. SIGGRAPH)*, 22:299–308, 1988.

[63] I. Han and B.J. Gilmore. Impact Analysis for Multiple Body Systems with Friction and Sliding Contact. *ASME Journal of Mechanical Design*, 115:412–422, September 1993.

[64] B. Hannaford and R.J. Anderson. Experimental and Simulation Studies of Hard Contact Force Reflecting Teleoperation. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, pages 24–29, Scottsdale, AZ, May 1988.

[65] B. Hannaford and J. H. Ryu. Time-Domain Passivity of Haptic Interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, February 2002.

[66] E.J. Haug, S.C. Wu, and S.M. Yang. Dynamics of Mechanical Systems with Coulomb Friction, Stiction, Impact and Constraint Addition-Deletion - I: Theory. *Mechanism and Machine Theory*, 21(5):401–406, 1986.

[67] V. Hayward and B. Armstrong. A new computational model of friction applied to haptic rendering. In P. Corke and J. Trevelyan, editors, *Experimental Robotics VI. Lecture Notes in Control and Information Sciences*, volume LNCS 250, pages 404–412. Springer: New-York, 2000.

[68] N. Hogan. Impedance Control: An Approach to Manipulation. Theory, Implementaion, Application. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, 107:1–24, March 1985.

[69] N. Hogan. Controlling Impedance at the Man/Machine Interface. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation*, pages 1626–1631, 1989.

[70] Y. Hurmuzlu and D.B. Marghitu. Three Dimoensional Rigid Body Collisions with Multiple Contact Points. *ASME Journal of Applied Mechanics*, 62:725–732, 1995.

[71] J. D. Hwang, M. D. Williams, and G. Niemeyer. Toward Event-Based Haptics: Rendering Contact Using Open-Loop Force Pulses. In *The 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 14–21, Chicago, IL, March 2004.

[72] K.W. Jablokow and C.S. Bonaventura. Alternate Forms of the Operational Space Inertia Matrix for the Dynamic Simulation of Complex Robot Systems. In *Proceedings of the 2003 International Conference on Systems, Man, and Cybernetics*, pages 16–21, Washington, DC, October 2003.

[73] A. Jain. Unified Formulation of Dynamics for Serial Multibody Systems. *Journal of Guidance, Control, and Dynamics*, 14(3):531–542, 1991.

[74] A. Jain and G. Rodriguez. Diagonalized Lagrangian Robot Dynamics. *IEEE Transactions on Robotics and Automation*, 11(4):571–584, 1995.

[75] D.E. Johnson and P. Willemsen. Six Degree-of-Freedom Haptic Rendering of Complex Polygonal Models. In *11th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 229–235, Los Angeles, Ca, 2003.

[76] D.E. Johnson and P. Willemsen. Six Degree-of-Freedom Haptic Rendering of Complex Polygonal Models. In *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 18–23, Chicago, IL, 2004.

[77] D. Karnopp. Computer Simulation of Slip-Stick Friction in Mechanical Dynamic Systems. *Transactions of the ASME, Journal of Dynamic Systems, Measurement, and Control*, 107:100–103, March 1985.

[78] J.B. Keller. Impact with Friction. *ASME Journal of Applied Mechanics*, 53:1–4, 1986.

[79] A.J. Kelley and S.E. Salcudean. The Development of a Force Feedback Mouse and its Integration into a Graphical User Interface. In *Proceedings of the 1994 ASME International Mechanical Engineering Congress and Exposition*, volume DSC-55-1, pages 287–294, Chicago, IL, 1994.

[80] O. Khatib. A Unified Approach for Motion and Force Control of Robot Manipulators: The Operational Space Formulation. *IEEE Transactions on Robotics and Automation*, 3(1):43–53, 1987.

[81] O. Khatib. Inertial Properties in Robotic Manipulation: An Object-Level Framework. *International Journal of Robotics Research*, 13(1):19–36, 1995.

[82] S.S. Kim and M.J. Vanderploeg. QR Decomposition for State Space Representation of Constrained Mechanical Systems. *ASME Journal on Mechanism, Transmissions and Automation in Design*, 108:183–188, 1986.

[83] Y.J. Kim, M.C. Lin, and D. Manocha. Incremental Penetration Depth Estimation between Convex Polytopes Using Dual-Space Expansion. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):152–163, March/April 2004.

[84] Y.J. Kim, M.A. Otaduy, M.C. Lin, and D. Manocha. Six-Degree-of Freedom Haptic Display Using Localized Contact Computations. In *Tenth Symposium on Haptic Interfaces For Virtual Environment and Teleoperator Systems*, pages 209–216, Washington, DC, March 2002.

[85] Y.J. Kim, M.A. Otaduy, M.C. Lin, and D. Manocha. Six-Degree-of Freedom Haptic Display Using Incremental and Localized Contact Computations. *Presence: Teleoperators and Virtual Environments*, 12(3):277–295, June 2003.

[86] P.R. Kraus and V. Kumar. Compliant Contact Models for Rigid Body Collisions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1382–1387, Albuquerque, NM, 1997.

[87] H.M. Lankarani and P.E. Nikravesh. A Contact Force Model with Histeresis Damping for Impact Analysis of Multibody Systems. *ASME Journal of Mechanical Design*, 112:369–376, 1990.

[88] E. Larsen, S. Gottschalk, M.C. Lin, and D. Manocha. Fast Proximity Queries with Swept Sphere Volumes. Technical Report TR99-018, University of North Carolina, Chapel Hill, NC, 1999.

[89] D.A. Lawrence. Stability and Transparency in Bilateral Teleoperation. *IEEE Transactions on Robotics and Automation*, 9(5):624–637, 1993.

[90] D.A. Lawrence, L. Pao, A.M. Dougherty, M.A. Salada, and Y. Pavlou. Rate-Hardness: A New Performance Metric for Haptic Interfaces. *IEEE Trans Rob Autom*, 16(4):357–371, 2000.

[91] K.W. Lilly and D.E. Orin. Efficient Dynamic Simulation of Multiple Chain Robotic Mechanisms. *Journal of Dynamic Systems, Measurement, and Control*, 116:223–231, 1994.

[92] P. Lotstedt. Coulomb Friction in Two-Dimensional Rigid Body Systems. *Zeitschrift fur Angewandte Mathematik und Mechanik*, 61(12):605–615, 1981.

[93] P. Lotstedt. Numerical Simulation of Time Dependent Contact and Friction Problems in Rigid Body Mechanics. *SIAM Journal of Scientific Statistical Computing*, 42(2):281–296, 1982.

[94] P. Lotstedt. Numerical Simulation of Time-Dependent Contact and Friction in Rigid Body Mechanics. *SIAM Journal of Scientific Statistical Computing*, 5(2):370–393, 1984.

[95] S. Mahapatra and M. Zefran. Stable Haptic Interaction with Swistched Virtual Environments. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, Taipei, Taiwan, September 2003.

[96] M. Mahvash and V. Hayward. High Fidelity Haptic Synthesis of Contact With Deformable Bodies. *IEEE Computer Graphics and Applications*, 24(2):48–55, 2004.

[97] T. Maneerwan, B. Hannaford, D. Sorti, and M. Ganter. Haptic Rendering for Internal Content of an Implicit Object. In *ASME Winter Annual Meeting Haptics Symposium*, Nashville, TN, November 1999.

[98] N.K. Mani, E.J. Haug, and K.E. Atkinson. Application of Singular Value Decomposition for Analysis of Mechanical System Dynamics. *ASME Journal on Mechanism, Transmissions and Automation in Design*, 107:83–87, 1985.

[99] D.B. Marghitu and Y. Hurmuzlu. Three Dimensional Rigid Body Collisions with Multiple Contact Points. *The ASME Journal of Applied Mechanics*, 62(3):725–732, 1995.

[100] W.R. Mark, S.C. Randolph, M. Finch, J.M. Van Verth, and R.M. Taylor II. Adding Force Feedback to Graphics Systems: Issues and Solutions. In *Haptic Virtual Reality for Blind Computer Users, Assistive Technologies*, pages 92–99, 1998.

[101] M. Mason. Mechanics and Planning of Manipulator Pushing Operations. *International Journal of Robotics Research*, 5(3):53–71, 1885.

[102] M. Mason and Y. Wang. On the Inconsistency of Rigid-Body Frictional Planar Mechanics. In *IEEE 1988 International Conference on Robotics and Automation*, pages 524–528, Philadephia, PA, 1988.

[103] W.A. McNeely, K.D. Puterbaugh, and J.J. Troy. Six Degree-of-Freedom Haptic Rendering using Voxel Sampling. In *Proceedings of the 1999 AMC SIGGRAPH*, pages 92–99, 1999.

[104] A. Miller and H.I. Christensen. Implementation of Multi-rigid-body Dynamics within a Robotic Grasping Simulator. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2262–2268, Taipei, Taiwan, September 2003.

[105] B.E. Miller, J.E. Colgate, and R.A. Freeman. Guaranteed Stability of Haptic Systems with Nonlinear Virtual Environments. *IEEE Transactions on Robotics and Automation*, 16(2):712–719, December 2000.

[106] J.K. Mills and C.V. Nguyen. Robotic Manipulator Collisions: Modeling and Simulation. *The AMSE Jounral of Dynamic Systems,, Measurement, and Control*, 114:650–658, December 1992.

[107] B. Mirtich. *Impulse-Based Dynamic Simulation of Rigid Body Systems*. PhD thesis, University of California, Berkeley, CA, 1996.

[108] B. Mirtich. V-Clip: Fast and Robust Polyhedral Collision Detection. *ACM Transactions on Graphics)*, 17(3):177–208, 1998.

[109] B. Mirtich. Timewarp Rigid Body Simulation. In *Proceedings of the 2000 ACM SIGGRAPH*, New Orleans, LA, 193-200 2000.

[110] B. Mirtich and J. Canny. Impulse-based Dynamic Simulation. In K. Goldberg, P. Halperin, J.-C. Latombe, and R. Wilson, editors, *Workshop on Algorithmic Foundations of Robotics*, pages 407–418. A.K.Peters, Boston, MA., 1994.

[111] M. Moore and J. Wilhelms. Collision Detection and Response for Computer Animation. *Computer Graphics (Proceedings of ACM 1988 SIGGRAPH Conference)*, 22:289–298, 1988.

[112] J.J. Moreau. Unilateral Contact and Dry Friction in Finite Freedom Dynamics. In J.J. Moreau and P.D. Panagiotopoulos, editors, *Nonsmooth Mechanics and Applications. CISM Courses and Lectures*, volume 302, pages 1–82. International Centre for Mechanical Sciences, Springer-Verlag, 1988.

[113] A. Nahvi, J.M. Hollerbach, R. Freier, and D.D. Nelson. Display of Friction in Virtual Environments based on Human Finger Pad Characteristics. In *Proceedings of the ASME Dynamics Systems and Control Division*, volume DSC-64, pages 179–184, 1998.

[114] A. Nahvi, D.D. Nelson, J.M. Hollerbach, and D.E. Johnson. Haptic Manipulation of Virtual Mechanisms from Mechanical CAD Designs. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 375–380, Leueven, Belgium, May 1998.

[115] P.L. Nash. A New First Integral for a Binary Rigid Body Collision of Arbitrarily Short Duration. *Jounral of Mathematical Physics*, 40(6):2816–2829, 1999.

[116] N. Orlandea, M.A. Chace, and D.A. Calahan. A Sparsity-Oriented Approach to the Dynamic Analysis and Design of Mechanical Systems - Part I. *Journal of Engineering in Industry*, 99:773–784, 1977.

[117] M.A. Otaduy and M.C. Lin. Sensation Preserving Simplification for Haptic Rendering. *ACM Transactions on Graphics. Special issue: Proceedings of the ACM SIGGRAPH 2003*, 22(3):543–553, 2003.

[118] J.-S. Pang and J.C. Trinkle. Complementarity Formulations and Existence of Solutions of Multi-Body Contact Problems with Coulomb Friction. *Mathematical Programming*, 73:199–226, 1994.

[119] J.G. Park and G. Niemeyer. Haptic Rendering with Predictive Representation of Local Geometry. In *Proceedings of the 12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 331–338, Chicago, IL, 2004.

[120] F. Pffeifer and C. Glocker. *Multibody Dynamics with Unilateral Contacts.* Wiley, New York, 1996.

[121] J.C. Platt and A.H. Barr. Constraint Methods for Flexible Models. *Computer Graphics (Proc. SIGGRAPH)*, 22:279–288, 1988.

[122] D. Redon, A. Kheddar, and S. Coquillart. Gauss' Least Constraints Principle and Rigid Body Simulations. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 517–522, Washington, DC, 2002.

[123] M. Renz, C. Preusche, M. Potke, H.P. Kriegel, and G. Hirzinger. Stable Haptic Interaction with Virtual Environments Using an Adapted Voxmap-Pointshell Algorithm. In *Proceedings of the Eurohaptics Conference*, pages 149–154, 2001.

[124] C. Richard and M. Cutkosky. The Effects of Real and Computer Generated Friction on Human Performance in a Targeting Task. In *Proceedings of the ASME/IMECE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Nashville, TN, 2000.

[125] L.B. Rosenberg and B.D. Adelstein. Perceptual Decomposition of Virtual Haptic Surfaces. In *Proceedings of the 1993 IEEE Symposium on Research Frontiers in Virtual Reality*, pages 46–53, San Jose, CA, 1993.

[126] D. Ruspini and O. Khatib. Dynamic Models for Haptic Rendering Systems. In *Advances in Robot Kinematics: ARK98*, pages 523–532, Strobl/Salzburg, Austria, 1998.

[127] D. Ruspini and O. Khatib. Collision/Contact Models for Dynamic Simulation and Haptic Interaction. In *Proceedings of the 9th International Symposium on Robotics Research*, pages 185–195, Snowbird, UT, 1999.

[128] D. Ruspini and O. Khatib. Haptic Display for Human Interaction with Virtual Dynamic Environments. *Journal of Robotic Systems*, 18(2):769–783, 2001.

[129] D. Ruspini, K. Koralov, and O. Khatib. The Haptic Display of Complex Graphical Environments. In *Proceeding of the SIGGRAPH 97*, pages 345–352, Los Angeles, CA, 1997.

[130] S.E. Salcudean. Control for Teleoperation and Haptic Interfaces. In B Siciliano and K.P. Valavanis, editors, *Lecture Notes in Control and Information Sciences 230 - Control Problems in Robotics and Automation*, pages 51–65. Springer-Verlag, 1997.

[131] S.E. Salcudean and T.D. Vlaar. On the Emulation of Stiff Walls and Static Friction with a Magnetically Levitated Input/Output Device. *ASME J Dyn Meas Contr*, 119:127–132, March 1997.

[132] S.E. Salcudean, N.M. Wong, and R.l. Hollis. Design and Control of a Force-Reflecting Teleoperation System with Magnetically Levitated Master and Wrist. *IEEE Transactions on Automatic Control*, 6:844–858, December 1995.

[133] J.K. Salisbury and C. Tarr. Haptic Rendering of Surfaces Defined by Implicit Fuctions. In *Proceedings of the ASME Dynamics Systems and Control Division*, volume DSC-61, pages 61–67, 1997.

[134] H. Schmidl and V.J. Milenkovic. A Fast Impulsive Contact Suite for Rigid Body Simulation. *IEEE Transactions on Vizualization and Computer Graphics*, 10(2):189–197, March/April 2004.

[135] K. Shimoga. A Survey of Perceptual Feedback Issues in Dexterous Telemanipulation: Part I. Finger Force Feedback. In *Proceedings of the 1993 IEEE Annual Virtual Reality International Symposium*, pages 263–270, Seattle, WA, 1993.

[136] M.R. Sirouspour, S.P. DiMaio, S.E. Salcudean, P. Abolmaesumi, and C. Jones. Haptic Interface Control – Design Issues and Experiments with a Planar Device. In *Proc 2000 IEEE Int Conf Rob Autom*, pages 789–794, San Francisco, Ca, 2000.

[137] C.E. Smith. Predicting Rebounds using Rigid-Body Dynamics. *Journal of Applied Mechanics*, 58:754–758, September 1991.

[138] W. Son, K. Kim, and N.M. Amato. An Interactive Generalized Motion Simulator (GMS) in an Object-Oriented Framework. In *Proceedings of the IEEE Computer Animation 2000*, pages 176–182, Phyladelphia, PA, May 2000.

[139] W. Son, K. Kim, N.M. Amato, and J.C Trinkle. Interactive Dynamic Simulation using Haptic Interaction. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, pages 145–150, San Francisco, CA, May 2000.

[140] W. Son, K. Kim, N.M. Amato, and J.C Trinkle. A Generalized Framework for Interactive Dynamic Simulation for MultiRigid Bodies. *IEEE Transactions on Systmes, Man, and Cybernetics - Part B: Cybernetics*, 34(2):912–924, April 2004.

[141] W. Son, J.C Trinkle, and N.M. Amato. An Interactive Generalized Motion Simulator (I-GMS) in an Object-Oriented Framework. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 3789–3794, Seoul, Corea, May 2001.

[142] D.E. Stewart. Rigid Body Dynamics with Friction and Impact. *SIAM Review*, 42(1):3–29, 2000.

[143] D.E. Stewart and J.C. Trinkle. An Implicit Time-Stepping Scheme Rigid Body Dynamics with Inellastic Collisions and Coulomb Friction. *The International Journal of Numerical Methods in Engineering*, 39(15):2673–2691, 1996.

[144] D.E. Stewart and J.C. Trinkle. Convergence of a Time-Stepping Scheme for Rigid Body Dynamics and Resolution of Painleve's Paradoxes. *Archives of Rational Mechanics*, 1998.

[145] W.J. Stronge. Rigid Body Collisions with Friction. *Proceedings of the Royal Society of London*, A431:169–181, 1990.

[146] F.E. Udwadia and R.E. Kabala. Explicit Equations of Motion for Mechanical Systmes with Nonideal Constraints. *Transactions of the ASME¡ Journal of Applied Mechanics*, 68:462–467, May 2001.

[147] G. van der Bergen. Proximity Queries and Penetration Depth Computation on 3D Game Objects. In *Game Developers Conference*, 2001.

[148] A. van der Schaft. *L2-gain and Passivity Techniques in Nonlinear Control.* Springer, London, U.K., 2000.

[149] S. Vedula and D. Baraff. Force feedback in interactive dynamic simulation. In *Proceedings of the First PHANTOM Users Group Workshop*, pages 54–57, 1996.

[150] S. Walker and J.K. Salisbury. Large haptic topographic maps and the proxy graph algorithm. In *Proceedings of the ACM 2003 Symposium on Interactive 3D Graphics*, pages 83–92, 2003.

[151] Y. Wang and M.T. Mason. Two-Dimensional Rigid-Body Collisions with Friction. *ASME Journal of Applied Mechanics*, 59:635–642, 1992.

[152] A. Witkin and D. Baraff. Physically Based Modeling: Principles and Practice. SIGGRAPH '97 Course Notes.

[153] S.C. Wu, S.M. Yang, and E.J. Haug. Dynamics of Mechanical Systems with Coulomb Friction, Stiction, Impact and Constraint Addition-Deletion - II: Planar Systems. *Mechanism and Machine Theory*, 21(5):407–416, 1986.

[154] S.C. Wu, S.M. Yang, and E.J. Haug. Dynamics of Mechanical Systems with Coulomb Friction, Stiction, Impact and Constraint Addition-Deletion - III: Spatial Systems. *JSME International Journal, Series C: Mechanical Systems, Machine Elements and Manufacturing*, 43(5):387–393, June 2000.

[155] C.B. Zilles and J.K. Salisbury. A Constraint-based God Object Method for Haptic Display. In *ASME Haptic Interf Virt Envir Teleop Syst*, pages 146–150, Chicago, IL, 1994.

Appendix A

Proof of implicit elimination of constraint overdeterminancy

This appendix proves that, if $\mathcal{J}_c$ is row rank deficient, then:

$$\mathcal{J}_c^T \left(\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T\right)^{\dagger} \mathcal{J}_c = \mathcal{J}_n^T \left(\mathcal{J}_n \mathbf{D}^{-1} \mathcal{J}_n^T\right)^{-1} \mathcal{J}_n \tag{A.1}$$

where $\mathcal{J}_n$ is full row rank, and $\mathcal{J}_c^T = \begin{bmatrix} \mathcal{J}_n^T & \mathcal{J}_r^T \end{bmatrix}$.

Proof:

Let $\mathcal{J}_c$ be given by $\mathcal{J}_c^T = \begin{bmatrix} \mathcal{J}_n^T & \mathcal{J}_r^T \end{bmatrix}^T$, where $\mathcal{J}_n$ is full row rank, i.e., $rank(\mathcal{J}_n) = rank(\mathcal{J}_c) = n$, and $rank(\mathcal{J}_r) = r$. By elementary row operations:

$$\begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{A} & \mathbf{I}_r \end{bmatrix} \begin{bmatrix} \mathcal{J}_n \\ \mathcal{J}_r \end{bmatrix} = \begin{bmatrix} \mathcal{J}_n \\ \mathbf{0} \end{bmatrix}, \tag{A.2}$$

where $\mathbf{A} \in \mathcal{R}^{r \times n}$, and $\mathbf{I}_n$ and $\mathbf{I}_r$ are identity matrices of rank n and r, respectively.

Then:

$$\begin{aligned} \left(\mathcal{J}_c \mathbf{D}^{-1} \mathcal{J}_c^T\right)^{\dagger} &= \left(\begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{A} & \mathbf{I}_r \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{J}_n \\ \mathbf{0} \end{bmatrix} \mathbf{D}^{-1} \begin{bmatrix} \mathcal{J}_n^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & \mathbf{A}^T \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix}^{-T} \right)^{\dagger} = \\ &= \left(\begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ -\mathbf{A} & \mathbf{I}_r \end{bmatrix} \begin{bmatrix} \mathcal{J}_n \\ \mathbf{0} \end{bmatrix} \mathbf{D}^{-1} \begin{bmatrix} \mathcal{J}_n^T & \mathbf{0}^T \end{bmatrix} \begin{bmatrix} \mathbf{I}_n & -\mathbf{A}^T \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix} \right)^{\dagger} = \\ &= \begin{bmatrix} \mathbf{I}_n & \mathbf{A}^T \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix} \begin{bmatrix} \mathcal{J}_n \mathbf{D}^{-1} \mathcal{J}_n^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}^{\dagger} \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{A} & \mathbf{I}_r \end{bmatrix}. \end{aligned} \tag{A.3}$$

The last algebraic manipulation is based on the fact that:

$$\left(\mathbf{X}\mathbf{X}^T\right)^\dagger = \left(\mathbf{X}^T\right)^\dagger \mathbf{X}^\dagger. \tag{A.4}$$

To show that Equation (A.4) holds, let the SVD of $\mathbf{X}$ be given as $\mathbf{X} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$. Then:

$$\left(\mathbf{X}\mathbf{X}^T\right)^\dagger \quad = \quad \left(\mathbf{U}\boldsymbol{\Sigma}\boldsymbol{\Sigma}\mathbf{U}^T\right)^\dagger = \mathbf{U}\boldsymbol{\Sigma}'\boldsymbol{\Sigma}'\mathbf{U}^T = \mathbf{U}\boldsymbol{\Sigma}'\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}'\mathbf{U}^T = \left(\mathbf{X}^T\right)^\dagger \mathbf{X}^\dagger, \tag{A.5}$$

where:

$$\boldsymbol{\Sigma}' = \begin{bmatrix} \frac{1}{\Sigma_1} & \cdots & \cdots & \cdots & 0 \\ \vdots & & & & \\ 0 & \cdots & \frac{1}{\Sigma_n} & \cdots & 0 \\ 0 & \cdots & 0 & \cdots & 0 \end{bmatrix}, \tag{A.6}$$

and $\Sigma_1, \cdots, \Sigma_n$ are the singular values of $\mathbf{X}$.

$\mathcal{J}_c^T \left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^\dagger \mathcal{J}_c$ can now be computed using Equations (A.2) and (A.3):

$$\begin{aligned} \mathcal{J}_c^T \left(\mathcal{J}_c\mathbf{D}^{-1}\mathcal{J}_c^T\right)^\dagger \mathcal{J}_c \quad &= \quad \mathcal{J}_c^T \begin{bmatrix} \mathbf{I}_n & \mathbf{A}^T \\ \mathbf{0} & \mathbf{I}_r \end{bmatrix} \begin{bmatrix} \mathcal{J}_n\mathbf{D}^{-1}\mathcal{J}_n^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{I}_n & \mathbf{0} \\ \mathbf{A} & \mathbf{I}_r \end{bmatrix} \mathcal{J}_c = \\ &= \begin{bmatrix} \mathcal{J}_n^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \left(\mathcal{J}_n\mathbf{D}^{-1}\mathcal{J}_n^T\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathcal{J}_n \\ \mathbf{0} \end{bmatrix} = \\ &= \mathcal{J}_n^T \left(\mathcal{J}_n\mathbf{D}^{-1}\mathcal{J}_n^T\right)^{-1} \mathcal{J}_n. \end{aligned} \tag{A.7}$$

Appendix B

Weighted generalized inverses for linkage manipulation

This appendix briefly overviews weighted generalized inverses applied to robotics, following the in-depth discussion of their properties and their correct application in robotics introduced in [46].

B.1 Weighted generalized inverses of matrices

Consider a linear transformation $\mathbf{A}$ that maps a real vector space X with metric $\mathbf{M}_x$ into a real vector space U with metric $\mathbf{M}_u$:

$$\mathbf{u} = \mathbf{A}\mathbf{x}. \tag{B.1}$$

If $\mathbf{A}$ is singular, then a solution $\mathbf{x}_s$ to Equation (B.1) may be derived as:

$$\mathbf{x}_s = \mathbf{A}^{\#}\mathbf{u}, \tag{B.2}$$

where $\mathbf{A}^{\#}$ is called the weighted generalized inverse of $\mathbf{A}$ [20] and is computed by:

$$\mathbf{A}^{\#} = \mathbf{M}_x^{-1}\mathbf{C}^T\left(\mathbf{C}\mathbf{M}_x^{-1}\mathbf{C}^T\right)^{-1}\left(\mathbf{F}^T\mathbf{M}_u\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{M}_u. \tag{B.3}$$

In Equation (B.3), $\mathbf{A} = \mathbf{F}\mathbf{C}$ is a full rank factorization of $\mathbf{A}$, i.e., $\mathbf{F}$ is full row rank and $\mathbf{C}$ is full column rank. $\mathbf{A}^{\#}$ is a weighted generalized inverse of $\mathbf{A}$ because, for any symmetric positive definite weights $\mathbf{M}_x$ and $\mathbf{M}_u$, it satisfies the following properties [20]:

1. $\mathbf{A}\mathbf{A}^{\#}\mathbf{A} = \mathbf{A}$

2. $\mathbf{A}^{\#}\mathbf{A}\mathbf{A}^{\#} = \mathbf{A}^{\#}$

3. $\left(\mathbf{M}_u\mathbf{A}\mathbf{A}^{\#}\right)^T = \mathbf{M}_u\mathbf{A}\mathbf{A}^{\#}$

4. $\left(\mathbf{M}_x\mathbf{A}^{\#}\mathbf{A}\right)^T = \mathbf{M}_x\mathbf{A}^{\#}\mathbf{A}$

Additionally, $\mathbf{A}^{\#}$ is unique, $\left(\mathbf{A}^{\#}\right)^{\#} = \mathbf{A}$ and $\left(\mathbf{A}^{\#}\right)^T = \left(\mathbf{A}^T\right)^{\#}$ [20].

The weighted generalized inverse $\mathbf{A}^{\#}$ computes the minimum $\mathbf{M}_x$-norm:

$$|\mathbf{x}_s|^2_{\mathbf{M}_x} := < \mathbf{x}_s, \mathbf{M}_x\mathbf{x}_s > , \tag{B.4}$$

of all solutions that produce the minimum $\mathbf{M}_u$-least squares error:

$$|\mathbf{u} - \mathbf{A}\mathbf{x}_s|^2_{\mathbf{M}_u} := < \left(\mathbf{u} - \mathbf{A}\mathbf{x}_s\right), \mathbf{M}_u\left(\mathbf{u} - \mathbf{A}\mathbf{x}_s\right) > , \tag{B.5}$$

where $<, >$ denotes the inner product on a metric space.

As discussed in [46], the weighted generalized inverse is invariant to the choice of metric $\mathbf{M}_x$ on X if $\mathbf{A} = \mathbf{C}$ has full column rank:

$$\mathbf{A}^{\#} = \mathbf{M}_x^{-1}\mathbf{C}^T\left(\mathbf{C}\mathbf{M}_x^{-1}\mathbf{C}^T\right)^{-1} = \mathbf{M}_x^{-1}\mathbf{A}^T\left(\mathbf{A}\mathbf{M}_x^{-1}\mathbf{A}^T\right)^{-1} \tag{B.6}$$

and is invariant to the choice of metric $\mathbf{M}_u$ on U if $\mathbf{A} = \mathbf{F}$ is full row rank:

$$\mathbf{A}^{\#} = \left(\mathbf{F}^T\mathbf{M}_u\mathbf{F}\right)^{-1}\mathbf{F}^T\mathbf{M}_u = \left(\mathbf{A}^T\mathbf{M}_u\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{M}_u. \tag{B.7}$$

If $\mathbf{A}$ is invertible, then its weighted generalized inverse is invariant to both metrics:

$$\mathbf{A}^{\#} = \left(\mathbf{A}^T\mathbf{M}_u\mathbf{A}\right)^{-1}\mathbf{A}^T\mathbf{M}_u = \mathbf{A}^{-1}\mathbf{M}_u^{-1}\mathbf{A}^{-T}\mathbf{A}^T\mathbf{M}_u = \mathbf{A}^{-1}. \tag{B.8}$$

B.2 Weighted generalized inverses applied to robotics

A linear transformation $\mathbf{A}$ (that maps X into U) and its transpose $\mathbf{A}^T$ (that maps W into T):

$$\mathbf{u} = \mathbf{A}\mathbf{x} \tag{B.9}$$

$$\mathbf{t} = \mathbf{A}^T\mathbf{w} \tag{B.10}$$

and the scalar product:

$$P := < \mathbf{u}, \mathbf{v} > \tag{B.11}$$

are called a *dual system* [46] if all terms in P have the same physical units. The importance of dual systems lies in the fact that their solution is simplified by the fact that $\mathbf{M}_x^{-1}$ and $\mathbf{M}_u^{-1}$ are metrics on T and W whenever $\mathbf{M}_x$ and $\mathbf{M}_u$ and metrics on X and U. Consequently, the dual system may be solved using the weighted generalized inverse $\mathbf{A}^{\#}$ and its transpose $\left(\mathbf{A}^{\#}\right)^T$:

$$\begin{aligned} \mathbf{x} &= \mathbf{A}^{\#}\mathbf{u} && \text{(B.12)} \\ \mathbf{w} &= \left(\mathbf{A}^{\#}\right)^T \mathbf{t}. && \text{(B.13)} \end{aligned}$$

Until now, the discussion of linear transformations and their weighted generalized inverses has been general, but it has implicitly relied on the metric structure of X and U and the inner products defined on these spaces through the metrics $\mathbf{M}_x$ and $\mathbf{M}_u$. The relevance of this theory to robotics (and to virtual linkage manipulation) is apparent when considering suitable choices of metrics on the linear spaces of interest.

In robotics, the linear transformations of interest are the Jacobian matrix of a linkage $\mathbf{J}$ and its transpose $\mathbf{J}^T$. The linkage Jacobian maps the space Q of joint velocities to the space V of twists (rigid body velocities):

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}. \tag{B.14}$$

In Equation (B.14), $\dot{\mathbf{q}}$ is the vector of configuration (linear and angular joint) velocities of the linkage, and $\dot{\mathbf{x}} = \begin{pmatrix}\mathbf{v}^T & \boldsymbol{\omega}^T\end{pmatrix}^T$ is the twist at the point of interest, with $\mathbf{v} \in \mathcal{R}^3$ being the linear velocity at the point of interest and $\boldsymbol{\omega} \in \mathcal{R}^3$ being the angular velocity of the link on which the point of interest lies. The transpose of the linkage Jacobian maps the space of wrenches F to the space of configuration (joint) torques T:

$$\mathbf{T} = \mathbf{J}^T\mathbf{F}. \tag{B.15}$$

In Equation (B.15), $\mathbf{T}$ is the vector of configuration (joint) torques, and $\mathbf{F} = \begin{pmatrix}\mathbf{f}^T & \boldsymbol{\tau}^T\end{pmatrix}^T$ is the wrench at the point of interest, with $\mathbf{f} \in \mathcal{R}^3$ being the force at this point and $\boldsymbol{\tau} \in \mathcal{R}^3$ being the torque. According to the discussion above, $\mathbf{J}$, $\mathbf{J}^T$, and the scalar product (representing power):

$$P := < \dot{\mathbf{x}}, \mathbf{F} > \tag{B.16}$$

form a dual system, i.e., $\dot{\mathbf{x}}$ and $\mathbf{F}$ (and $\dot{\mathbf{q}}$ and $\mathbf{T}$) are said to be duals of each other. This means that finding suitable metrics on Q and V is sufficient for inverting both Equation (B.14) and Equation (B.15). Therefore, only the mapping of configuration velocities to twists is considered in the following.

When choosing metrics on these two spaces, two conditions must be satisfied in order for the metrics to have physical significance:

1. The inner products on the two spaces must have the same units in all the terms (i.e., they must be physically consistent).
2. The metric must be gauge invariant (i.e., invariant to coordinate, or rigid body, transformations).

As shown in [46], kinetic energy metrics satisfy these two properties.

This thesis is concerned with enabling realistic and unrestricted (i.e., from any user-selected link and through singularities) haptic manipulation of arbitrary virtual linkages. Therefore, two kinetic energy metrics are chosen to compute the weighted generalized inverse of the Jacobian of the virtual tool at the user's hand, according to:

$$\mathbf{J}_h^{\#} = \mathbf{D}_{vt}^{-1}\mathbf{C}^T \left(\mathbf{C}\mathbf{D}_{vt}^{-1}\mathbf{C}^T\right)^{-1} \left(\mathbf{F}\mathbf{M}_{hl}\mathbf{F}^T\right)^{-1} \mathbf{F}^T\mathbf{M}_{hl}. \tag{B.17}$$

In Equation (B.17), $\mathbf{D}_{vt}$ is the configuration space inertia of the virtual tool and $\mathbf{M}_{hl}$ is the inertia of the link held by the user, computed in the world coordinate frame. $\mathbf{J}_h^{\#}$ is used to represent the contact interactions of the virtual tool at the user's hand in a coordinate-invariant manner when the user holds a link whose motion is restricted by the linkage to which it belongs. Example such interactions include a user rotating a virtual crank or opening a virtual door.

Note that, if the Jacobian of the virtual tool at the user's hand is full column rank, i.e., $\mathbf{J}_h = \mathbf{C}$, its weighted generalized inverse reduces to the dynamically consistent inverse of the Jacobian [80]:

$$\mathbf{J}_h^{\#} = \mathbf{D}^{-1}\mathbf{J}_h^T \left(\mathbf{J}_h\mathbf{D}^{-1}\mathbf{J}_h^T\right)^{-1}. \tag{B.18}$$

When the Jacobian looses rank, i.e., when the linkage topology restricts the rigid body motion at the user's hand, the inertia matrix of the link held by the user $\mathbf{M}_{hl}$ selects the rigid body motion that best approximates, in the $\mathbf{M}_{hl}$-least-squares sense, the rigid body motions not in the range of $\mathbf{J}_h$.

Appendix C

Equality of null spaces of $\mathbf{\Lambda}_h^{-1}$ and $\mathbf{J}_h^T$

This appendix proves that the SVD of the virtual tool Jacobian computed at the user's hand $\mathbf{J}_h$ can be used to derive the singular directions of the inverse of the virtual linkage operational space inertia at the user's hand $\mathbf{\Lambda}_h^{-1}$.

Let the SVD of $\mathbf{J}_h$ be given by:

$$\mathbf{J}_{h_{6\times 6}} = \mathbf{U}_{6\times 6}\mathbf{\Sigma}_{6\times d}\mathbf{V}^T_{d\times d}. \tag{C.1}$$

Then:

$$\mathbf{\Lambda}^{-1}_{h_{6\times 6}} \quad = \quad \mathbf{U}_{6\times 6}\mathbf{\Sigma}_{6\times d}\mathbf{V}^T_{d\times d}\mathbf{D}^{-1}_{vt_{d\times d}}\mathbf{V}_{d\times d}\mathbf{\Sigma}^T_{d\times 6}\mathbf{U}^T_{6\times 6}. \tag{C.2}$$

Since $\mathbf{D}_{vt}^{-1}$ is symmetric positive definite, it can be reduced to the diagonal form $\mathbf{A}$ through a suitable rotation $\mathbf{R}$:

$$\mathbf{A} = \begin{pmatrix} a_1 & 0 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & a_d \end{pmatrix} = \mathbf{R}\mathbf{D}_{vt}^{-1}\mathbf{R}^T. \tag{C.3}$$

Moreover, $\mathbf{Y} = \mathbf{RV} = (\mathbf{y}_1 \quad \dots \quad \mathbf{y}_d)$ is orthogonal (since $\mathbf{V}^T\mathbf{R}^T\mathbf{RV} = \mathbf{I}$), and the product $\mathbf{V}^T\mathbf{D}_{vt}^{-1}\mathbf{V}$ is diagonal and equal to $\mathbf{A}$:

$$\begin{aligned} \mathbf{V}^T\mathbf{D}_{vt}^{-1}\mathbf{V} &= \mathbf{V}^T\mathbf{R}^T\mathbf{ARV} = \\ &= \begin{pmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_d^T \end{pmatrix} \begin{pmatrix} a_1 & 0 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & a_d \end{pmatrix} \begin{pmatrix} \mathbf{y}_1 & \dots & \mathbf{y}_d \end{pmatrix} = \\ &= \begin{pmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_d^T \end{pmatrix} \begin{pmatrix} a_1\mathbf{y}_1 & \dots & a_d\mathbf{y}_d \end{pmatrix} = \mathbf{A}. \end{aligned} \tag{C.4}$$

Furthermore, let $rank\left(\mathbf{J}_h\right) = r$, i.e.,:

$$\boldsymbol{\Sigma}_{6\times d} = \begin{bmatrix} \boldsymbol{\Sigma}'_{r\times r} & \mathbf{0}_{r\times(d-r)} \\ \mathbf{0}_{(6-r)\times r} & \mathbf{0}_{(6-r)\times(d-r)} \end{bmatrix}, \tag{C.5}$$

where $\boldsymbol{\Sigma}^{'}_{r\times r}$ is the r-dimensional diagonal matrix having the r non-zero singular values $\Sigma_1,\ldots,\Sigma_r$ of $\mathbf{J}_h$ on its main diagonal. Then, from Equation (C.2) and Equations (C.4) and (C.5), it follows that:

$$\begin{aligned} \boldsymbol{\Sigma}_{\Lambda_{6\times6}} &= \boldsymbol{\Sigma}_{6\times d}\mathbf{V}^T_{d\times d}\mathbf{D}^{-1}_{vt_{d\times d}}\mathbf{V}_{d\times d}\boldsymbol{\Sigma}^T_{d\times 6} = \\ &= \begin{bmatrix} \boldsymbol{\Sigma}'_{r\times r}\mathbf{A}_{r_{r\times r}}\boldsymbol{\Sigma}'_{r\times r} & \mathbf{0}_{r\times(d-r)} \\ \mathbf{0}_{(d-r)\times r} & \mathbf{0}_{(d-r)\times(d-r)} \end{bmatrix}. \end{aligned} \tag{C.6}$$

In Equation (C.6), $\mathbf{A}_{r_{r\times r}}$ is the diagonal matrix having $a_1,\ldots,a_r$ on its main diagonal, and:

$$\boldsymbol{\Sigma}'\mathbf{A}\boldsymbol{\Sigma}' = \begin{bmatrix} a_1\Sigma_1^2 & \ldots & 0 \\ \vdots & & \\ 0 & \ldots & a_r\Sigma_r^2 \end{bmatrix}. \tag{C.7}$$

Finally, after substitution from Equation (C.7), Equation (C.2) becomes:

$$\boldsymbol{\Lambda}^{-1}_{h_{6\times6}} = \mathbf{U}_{6\times6}\boldsymbol{\Sigma}_{\Lambda_{6\times6}}\mathbf{U}^T_{6\times6}, \tag{C.8}$$

with $\mathbf{U}_{6\times6}$ orthogonal, and $\boldsymbol{\Sigma}_{\Lambda_{6\times6}}$ diagonal and having the first r diagonal elements non-zero. In other words, Equation (C.8) gives the SVD of $\boldsymbol{\Lambda}_h^{-1}$.

Appendix D

The dynamics and control of the planar haptic interface

The planar haptic interface used in the experiments described in this thesis is shown in Figure D.1. The interface comprises two identical pantograph mechanisms mounted in parallel. As depicted in Figure D.2, each pantograph consists of four carbon fibre links and five rotational joints. The pantograph endpoints are connected through a link that freely rotates with respect to each mechanism. This mechanical arrangement endows the haptic interface with three DOFs, translation along any horizontal axis and unlimited rotation about the vertical axis.

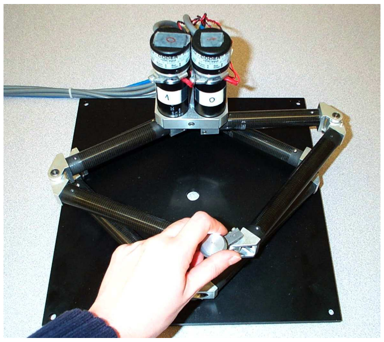


Figure D.1: The twin pantograph planar haptic interface.

D.1 Interface actuation and dynamics

Joint angles θ_1 and θ_2 are measured by digital encoders with a resolution of 0.0225 degrees (16,000 counts per revolution). These joints are actuated by 90W DC motors and are considered torque sources. All other joints are passive. The four idential motors that actuate the haptic interface are driven by constant current sources

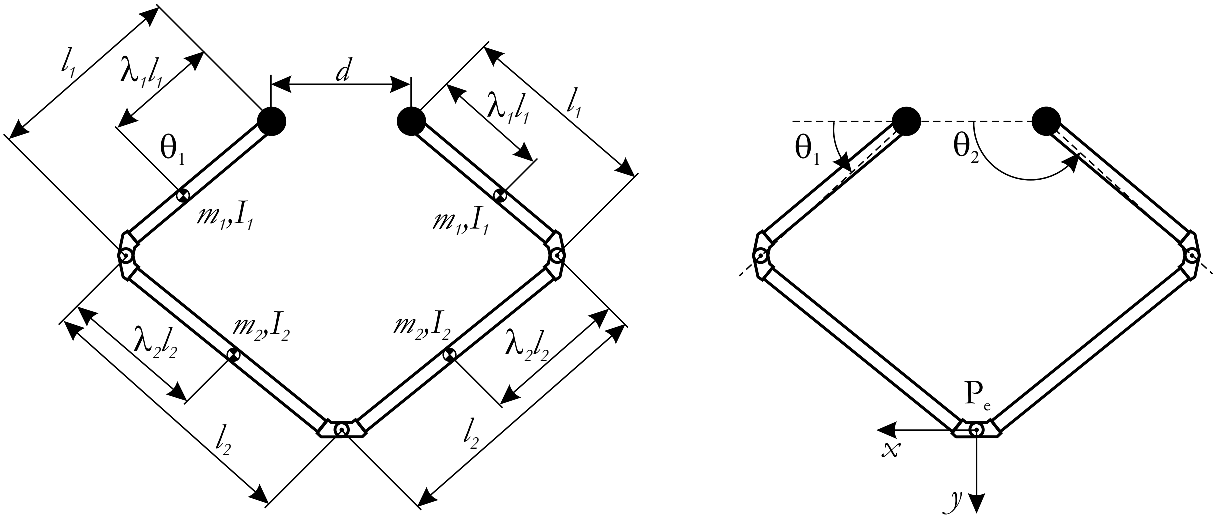


Figure D.2: The parameters of one pantograph linkage. The base joints are shown in black. The pantograph endpoint P_e is the joint between the two not grounded links.

and produce torques T proportional to the armature current i_a through the motor torque constant k_t:

$$T = k_t i_a. \tag{D.1}$$

The motors have a motor torque constant $k_t = 0.0525\text{NmA}^{-1}$, a continuous current rating of 2A, and a peak current rating exceeding 10A. They provide a peak force of 23N and a peak torque of 115Ncm at the ahndle. Peak accelerations of 6g have been obtained at half the maximum coil current.

Joint velocities $\dot{\theta}_1$ and $\dot{\theta}_2$ and user-applied wrenches $\mathbf{F}_h$ are estimated from joint angle measurements and applied motor torques using a force observer [60] and accurate interface dynamics [136]. In terms of the actuated joint variables $\theta = (\theta_1 \quad \theta_2)^T$, the configuration space dynamics of each pantograph are derived as [136]:

$$\mathbf{D}_p(\theta)\,\ddot{\theta} + \mathbf{C}_p(\theta, \dot{\theta})\dot{\theta} = \mathbf{T}_p - \mathbf{J}_p^T \mathbf{f}_p, \tag{D.2}$$

where $\mathbf{D}_p$ and $\mathbf{C}_p$ are the configuration space inertia and Christoffel matrices of one pantograph, respectively, $\mathbf{T}_p$ is the vector of applied actuator torques, $\mathbf{J}_p$ is the pantograph Jacobian computed at its endpoint, and $\mathbf{f}_p$ is the external force applied at its endpoint. The configuration space dynamics of the individual pantographs are combined into the operational space dynamics of the haptic interface [136]:

$$\mathbf{M}_d \ddot{\mathbf{x}}_h + \mathbf{C}_d \dot{\mathbf{x}}_h = \mathbf{F}_h + \mathbf{J}_e^T \mathbf{T} = \mathbf{F}_h + \mathbf{u}. \tag{D.3}$$

In Equation (D.3), $\ddot{\mathbf{x}}_h$, $\dot{\mathbf{x}}_h$, $\mathbf{x}_h$ are the body acceleration, velocity, and position of the user's hand (i.e., device handle), respectively, $\mathbf{M}_d$ and $\mathbf{C}_d$ are the operational space inertia and Christoffel matrices of the haptic interface, respectively, $\mathbf{F}_h$ is the user-applied wrench, $\mathbf{J}_e$ is the Jacobian matrix of the interface, $\mathbf{T}$ is the vector of applied actuator torques, and $\mathbf{u}$ is the operational space control signal.

Since the haptic interface has four motors and three DOFs, the actuation redundancy is used to minimize the force along the link connecting the two pantograph endpoints:

$$\mathbf{T} = \mathbf{J}_e^T \left(\mathbf{J}_o^T\right)^\dagger \mathbf{u}, \tag{D.4}$$

where $\mathbf{J}_o^\dagger = \mathbf{J}_o \left(\mathbf{J}_o^T \mathbf{J}_o\right)^{-1}$ is the right pseudo-inverse of $\mathbf{J}_o^T$. In turn, $\mathbf{J}_o^T$ maps the forces applied by the endpoints of the two pantograph mechanisms to the wrench applied by the haptic interface:

$$\mathbf{u} = \mathbf{J}_o^T \begin{pmatrix} f_{x1} \\ f_{y1} \\ f_{x2} \\ f_{y2} \end{pmatrix}. \tag{D.5}$$

D.2 Interface control

The interface is controlled by considering that the haptic interaction within virtual environments is a teleoperation task. According to this view, the haptic interface is the master and the virtual tool is the slave. High performance interaction [89,130] is achieved by controlling the haptic device using a four channel teleoperation architecture [89]. The advantage of the four channel architecture is that it allows the virtual environment to be designed independently from the haptic interface control.

The four channel architecture is illustrated in Figure D.3 for one DOF interaction. In this figure, f_h and f_{env} are the user-applied and the environment force, respectively, $\dot{x}_h$ and $\dot{x}_{h_{vt}}$ are the user's hand and the virtual tool velocities, respectively, and Z_h, Z_d, Z_{vt}, and Z_{env} are the impedances of the user's hand, the device, the virtual tool, and the virtual environment, respectively. Furthermore, C_1 and C_4 are the position channel controllers, C_2 and C_3 are the force channel controllers, and C_d and C_{vt} are the position controllers of the local device and the virtual tool, respectively.

Transparent interaction is difficult to achieve if the master and the slave are kinematically and dynamically dissimilar [130]. Therefore, an impedance controller shapes the impedance of the haptic device to match the desired impedance of the virtual tool, i.e., $Z_d = Z_{vt}$. Impedance shaping is performed according to Equation (5.21).

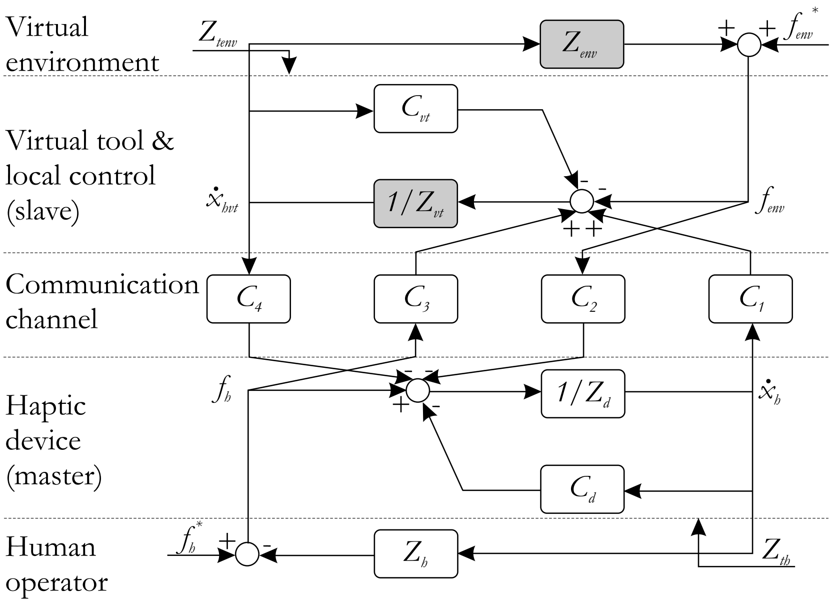


Figure D.3: The four channel architecture for 1DOF haptic interaction within virtual environments. Simulated blocks are shaded.

In this thesis, the following controller parameters were used:

$$\begin{aligned}
C_d &= 35 + \tfrac{100}{s} \\
C_{vt} &= 35 + \tfrac{100}{s} + B_{adp} \\
C_1 &= C_{vt} \\
C_2 &= 1 \\
C_3 &= 1 \\
C_4 &= -C_d
\end{aligned} \tag{D.6}$$

The adaptive damping $B_{adp} = K_b f_{env} + B_{min}$ is added to C_{vt} in order to reduce chattering ($K_b = 60$s/m, $B_{min} = 0$kg/s) [132].