creative commons
COMMONS DEED

저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:

저작자표시. 귀하는 원저작자를 표시하여야 합니다.

비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.

변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 이용허락규약(Legal Code)을 이해하기 쉽게 요약한 것입니다.

Disclaimer

dCollection

공학박사학위논문

수동성기반 다중접촉 햅틱렌더링 및 인터렉티브 시뮬레이션 프레임워크 개발

Passivity-Based Haptic Rendering and Interactive Simulation Framework with Multi-Point Contacts

2019 년 2 월

서울대학교 대학원

기계항공공학부

김 명 신

수동성기반 다중접촉 햅틱렌더링 및 인터렉티브 시뮬레이션 프레임워크 개발

Passivity-Based Haptic Rendering and Interactive Simulation Framework with Multi-Point Contacts

지도교수 이 동 준

이 논문을 공학박사 학위논문으로 제출함

2018년 10월

서울대학교 대학원

기계항공공학부

김 명 신

김명신의 박사 학위논문을 인준함

2018년 12월

위 원 장 박 종 우 (인)

부위원장 이 동 준 (인)

위 원 조 맹 효 (인)

위 원 박 재 흥 (인)

위 원 박 재 영 (인)

Abstract

In this thesis, we propose a haptic rendering and interactive simulation framework based on **passive midpoint integrator (PMI)** which can enforce discrete-time passivity of the simulation effectively in practice while retaining real-time interactivity from its being non-iterative. We derive this PMI simulation for mechanical systems both in generalized coordinates (i.e., in $\Re^n$) and also in maximal coordinates (i.e., in SE(3)), with some potential actions as well to implement joint articulation, constraints, compliance, etc. We also formulate both constraint-based contact solving and penalty-based contact solving for the PMI simulation, to resolve multi-point contact stably and interactively. To address/improve numerical stability and accuracy when too many contacts are engaged, we also propose a data-driven learning-based contact clustering, which consists of the multilayer perceptron (MLP) network and the constraint-based optimization contact solver.

We manifest the advantage of the proposed PMI-based simulation framework with some illustrative examples: 1) haptic rendering of peg-in-hole task, where very light/stiff articulated objects can be simulated with multi-point contact; 2) haptic interaction with flexible beam, where marginally-stable/lossless behavior (i.e., vibration) can be stably emulated; 3) under-actuated tendon-driven hand grasping, where mixed maximal-generalized coordinates are used with very light/stiff fingers; and 4) multiuser haptic interaction under delayed communication network, where peer-to-peer control architecture and passive consensus control are employed. We also verify the performance of the proposed data-driven/learning-based contact clustering framework, against the experimental contact data, compare it with other techniques/simulators, and show that there is significant (or meaningful) enhancement in the accuracy.

Keywords: Passivity, haptic rendering, interactive simulation, midpoint integration, multi-point contacts, data-driven contact.

Student Number: 2012-20651

Contents

1 Introduction **1**

1.1 Research Objectives and Background 1

1.2 Outline 5

2 Preliminary **7**

2.1 Mechanical Systems and Passivity 7

2.2 Discrete-Time Passivity and NPMI for Linear Mechanical Systems . . 11

2.2.1 Comparison of EEI, SEI, IEI and PMI 13

3 Development of Passive Midpoint Integrator **17**

3.1 Passive Midpoint Integrator in Generalized Coordinate 17

3.1.1 PMI of Nonlinear Lagrange Dynamics 17

3.1.2 Comparison of PMI with Symplectic Integrators 22

3.2 Passive Midpoint Integrator in Maximal Coordinate 26

3.2.1 PMI of Rigid Body in SE(3) 26

3.2.2 PMI-Based Rendering of Articulated Rigid Bodies 31

3.2.3 Improving Real-Time Interactivity 35

3.3 PMI-Based Multi-Point Contact Simulation 37

3.3.1 Constraint-Based Contact Solving 38

3.3.2 Penalty-Based Contact Solving 45

3.4 Passive Virtual Coupling for PMI Simulation 47

3.4.1 Improving Transparency of Virtual Coupling 52

3.4.2 Performance Evaluation of Transparent Virtual Coupling . . . 59

3.5 Illustrative Examples of PMI-Based Haptic Rendering 69
3.5.1 Peg-in-Hole Task 69
3.5.2 Haptic Interaction with Flexible Beam 72
3.5.3 Under-Actuated Tendon-Driven Hand Grasping 76

4 Contact Clustering and Data-Driven Approach 81
4.1 Motivation 81
4.1.1 K-Means Clustering 83
4.2 Data-Driven Contact Clustering 84
4.3 Data Acquisition and Network Training 86
4.4 Performance Evaluation of Data-Driven Clustering 88
4.4.1 Comparison with Open-Source Robot Simulators 91

5 Application to Multiuser Haptic Interaction with Wearable Haptic Interface 95
5.1 Design of Peer-to-Peer Control Architecture with Consensus Control . 96
5.2 Experimental Results and Performance Comparison 101

6 Conclusion and Future Works 105

List of Figures

2-1 Feedback interconnection of passive systems are stable by the passivity theorem [1]. Thus, we can make human-in-the loop simulation stable by passively implementing discrete-time simulation and interfacing between them (i.e., haptic rendering and control). 10

2-2 Simple harmonic oscillation simulation with EEI, SEI, IEI and PMI: $m = 0.01$[kg], $k = 100$[N/m], $T_k = 0.001$[sec]. 15

2-3 Simple harmonic oscillation simulation with EEI, SEI, IEI and PMI: $m = 0.00001$[kg], $k = 100$[N/m], $T_k = 0.001$[sec]. 15

3-1 Simulation of two serial-link robot arm with linear joint springs with SEI and PMI: the SEI is symplectic, yet, not exactly energy-preserving, whereas the PMI is energy-preserving, yet, not exactly symplectic. . 23

3-2 Total energy of rotational dynamics (3.31) with SO(3) spring (3.43) with nearly perfect energy conservation and stable/sustained oscillation in SO(3): with $J = 0.001 \cdot I$[kg m^2], $k_w = 1$ [Nm/rad] and $T_k = 0.01$ [sec] starting from $\phi_o = 0.9\pi$ along the direction of $\frac{1}{\sqrt{6}}(1, 2, 1)$. 30

3-3 Multi-fingered hand rendered as articulated multiple rigid bodies with joints implemented by point constraint springs. 31

3-4 Contact configuration of single rigid body contact: $\vec{n}_i$ is the normal direction, p_i^c the contact point on the object boundary, $p_{w_i}^c$ the contact point on the wall boundary, and δ_i is the penetration. Constraint-based method (left) enforces no penetration constraint while penalty-based method (right) allows some penetration. 38

3-5 Snapshots of rigid box falling on rigid level surface: it bounces back and forth under PMI-LCP (white box), whereas completely plastically collides under SEI (black box) even with the same parameter setting. 40

3-6 Profiles of total energy and height with PMI-LCP (3.79)-(3.80): elastic-normal and dissipative-tangential slips contact attained with total energy remained constant during flight, whereas strictly decreased during contact. Sudden changes of H around 4-4.2[sec] are due to collision with flipping. 41

3-7 Profiles of total energy and height under PMI-LCP (3.79)-(3.80) with $\hat{V}_k^a$ replaced by V_{k+1} for (3.75)-(3.76): completely plastic collision attained as in Fig. 3-8 under SEI. 42

3-8 Profiles of total energy and height with SEI: although the setting is the same as Fig. 3-6, completely plastic collision is attained (see also Fig. 3-5). 43

3-9 Block diagram of haptic interaction, where human users interact with virtual environments via the cascade of haptic device, virtual coupling, and virtual proxy. 49

3-10 Degradation of transparency of standard virtual coupling during contact task due to device-proxy position deviation and unscaled inertias of the device and the proxy. Our proposed virtual coupling aims to render the haptic device and the virtual proxy as one single rigid tool with perfect device-proxy coordination and zero apparent device-proxy inertia via inertia scaling. 51

3-11 Comparison of the standard virtual coupling (3.97)-(3.98) with the proposed virtual coupling (3.120)-(3.121) during 3-DOF haptic interaction with intermittent contacts, the first three on the middle of the proximal link and the last three on the end-effector. Notice our proposed virtual coupling maintains the device-proxy coordination even during the contacts, whereas the standard virtual coupling does not. 61

3-12 Device-proxy coordination performance of the standard virtual coupling and the proposed virtual coupling. Our proposed virtual coupling can maintain the coordination regardless of contact force, whereas coordination error of the standard virtual coupling is proportional to contact force. 62

3-13 Comparison of human forcing to generate spatially/temporally-similar free motion with three different inertia scaling factors. Notice that the human force gets smaller with $\eta = 0.6$ (i.e., inertia scaled-down) and larger with $\eta = 5$ (i.e., inertia scaled-up) as compared to $\eta = 1$ (i.e., no inertia scaling). 63

3-14 Controller passivity E_c of the proposed virtual coupling and the standard virtual coupling (averaged for 10 trials with minimum/maximum indicated too). Notice that inertia scaling down (i.e., $\eta < 1$) reduces "margin" of passivity, whereas inertia scaling up (i.e., $\eta > 1$) increases this margin, implying that our proposed virtual coupling is passive if η is not so small. 64

3-15 Frequency content of human motion during the experiments in Fig. 3-14 averaged over time. Note that the frequency content is mostly limited below 5Hz. 67

3-16 Snapshots of peg-in-hole task with two multi-fingered hands, each modeled as articulated rigid bodies as shown in Fig. 3-3 and making contact with the objects via friction contact simulated by PMI-LCP of Sec. 3.3.1. 70

3-17 Finger-tip contact force during the peg-in-hole task of Fig. 3-16 expressed in the finger-tip coordinate frame (with z-axis being the contact normal direction). 71

3-18 Snapshots of PMI-based simulation of undamped flexible beam harmonic oscillation. 72

3-19 Profiles of kinetic, potential and total energies of the PMI-based simulation of undamped flexible beam vibration in Fig. 3-18: note the exact conservation of total energy during the fast oscillations. 73

3-20 Profiles of kinetic, potential and total energies of the IEI-based simulation of undamped flexible beam vibration in Fig. 3-18 with linearized inertia matrix: note the total energy quickly decays due to dissipativity of the IEI, although the same initial conditions as that of Fig. 3-19 are used. 74

3-21 Flexible beam (in generalized coordinates) is connected to rigid body (in maximal coordinates) via spring, which is in turn connected to haptic device via virtual coupling. 75

3-22 Haptic feedback from the flexible beam to the haptic device through virtual coupling connected to the rigid box. 75

3-23 The iRobot-Harvard-Yale (iHY) Hand (left), the three fingered virtual UATD hand used in the simulation (middle), and the tendon routing of each finger (right.) 76

3-24 Snapshots of adaptive power-grasping of virtual rigid ball by the three-fingered UATD hand, with each finger having tendon-routing as depicted in Fig. 3-23, with the three fingers simultaneously controlled by one-DOF of the haptic device. 78

3-25 Haptic feedback of the one-DOF (i.e., z-directional) contact force of the adaptive power-grasping of the UATD hand in Fig. 3-24. 78

4-1 Simulation process with multi-point contacts. Contact clustering is employed for real-time fast and numerically stable contact simulation. 82

4-2 Architecture of our data-driven contact clustering: MLP network for clustering and constraint-based contact solver. The input of the MLP network is dynamics information of $\Re^{24}$ with single depth hidden layer. 85

4-3 Experiment setup consists of one KUKA robot manipulator, two ATI gamma sensors, and aluminium peg and plate. Vertically placed peg collides to the plate with different velocity for the first contact scenario (left) and inclined peg collides with different velocity for the second contact scenario (right). 87

4-4 Contact simulation performance: contact force of (4.1)-(4.4). For the surface contact (left), contact force with the machine learning-based clustering (w/ MLC, blue), with the k-means clustering (w/ KMC, green), and with the k-means clustering and warm starting (w/ KMCW, yellow) are presented. For the edge contact (right), contact force with the machine learning-based clustering (w/ MLC, blue), and without machine learning-based clustering (w/o MLC, green) are presented. 88

4-5 Illustrative example of learning-based clustering: learning-based clustering produces different contact points compared to standard k-means clustering (left) or normal (right) depending on X_k (e.g, external force and colliding velocity). 89

4-6 Configuration of V-REP robot simulation. Same real KUKA robot controller is used to control the virtual KUKA robot via remote API. 90

4-7 Comparison of surface contact simulations done by V-REP and experiment. We also plot the result of our proposed data-driven contact result. 91

4-8 Comparison of edge contact simulations done by V-REP and experiment. Black line represents the result of our proposed data-driven learning-based contact. 92

5-1 Peer-to-peer control architecture for the collaborative peg-in-hole haptic interaction among two remote users. 97

5-2 Wearable cutaneous haptic interface consists of two modules, namely: finger tracking module (FTM) for multi-DOF finger/hand motion estimation and cutaneous haptic module (CHM) foe 3-DOF contact force feedback on the fingertips. 100

5-3 Snapshot of collaborative shared virtual rigid body manipulation among two remote users implemented via the PMI simulation (left) and the open-source physics engine Bullet (right). The PMI-based interaction system can maintain its stability with a communication delay larger than 100[ms] under $K_c = 100$[N/cm]. On the other hand, the Bullet-based interaction system becomes unstable with a communication delay larger than 50[ms] under $K_c = 75$[N/cm]. 102

Chapter 1

Introduction

1.1 Research Objectives and Background

The goal of this thesis is to provide a passivity-based framework for haptic rendering and interactive simulation of dynamical systems which enables users to physically and/or graphically interact with a virtual dynamical environment. Due to their interactive nature, these haptic rendering and interactive simulation have two essential requirements, real-time update rate and robust stability. The real-time update rate is required to emulate a simultaneous and responsive mechanical and visual feedback. For example, the haptic rendering, for which some avatar of the haptic device (e.g., god-object [2], virtual proxy [3, 4]) and the virtual objects are simulated together, typically requires rather fast update rate (e.g., 500Hz-1kHz) for a continuous force feedback, whereas the interactive simulation, which typically deals with virtual objects and entities more complex than that of haptic rendering [5, 6] , requires slower update rate (e.g., 10-50Hz), which is yet fast enough for users to perceive continuous visual feedback. On the other hand, the interactive simulation and haptic rendering should maintain its stability robustly even under engagement of human user or intermittent contacts. However, guaranteeing stability based on dynamics analysis is hard for this interactive system, since dynamics of human interaction is hard to modelling/-analysing and the update rate changes in real-time simulation due to the intermittent contacts and/or human interaction. While considering the two requirements and their

difficulties, in this thesis, we propose a haptic rendering and interactive simulation framework which is based on 1) the passive midpoint integrator (PMI), and 2) the PMI-based multi-point contact handling including the contact clustering.

First of all, for the development of PMI, the core of any frameworks in haptic rendering and interactive simulation of dynamical systems [1] is the time integration. The majority of state-of-the-art frameworks adopts symplectic Euler integration (SEI [9], also often referred to as semi-implicit Euler in the field of haptics and computer animation: e.g., [10, 11, 12]) or implicit Euler integration (IEI: e.g., [13, 14, 6, 15]). Widely-used commercial physics engines and interactive simulation tools also rely on the SEI (e.g., CHAI-3D [16] based on ODE [17], VREP [18] based on ODE, Bullet [19], or Vortex [20]). The implicit Euler integration, in its exact form, is known to be (strictly) stable (e.g., [21, 9]), allowing for wide range of parameters (e.g., lighter mass with stiffer spring for transparency) and longer integration interval (i.e., faster simulation), yet, typically implemented only approximately for interactively-fast running (e.g., linearization [13, 14], explicitization [22, 23]). This IEI is also in general strictly dissipative, thus, cannot simulate lossless or energy-conserving behaviors (e.g., vibrating flexible object). The symplectic Euler integration is usually simpler to implement (thus, faster) than the IEI due to its semi-implicit nature. This SEI, however, is known to be only conditionally stable (e.g., [9]), which turns out to be fairly detrimental in practice, as one cannot freely tune and explore parameters of virtual objects to achieve (possibly changing) target performance independently from other parts of the virtual world.

On the other hand, passivity has been one of the most powerful tools in the field of haptics, telerobotics, and interactive robotics to enforce stability among dynamic entities coupled via feedback interconnection (e.g., between robots and humans, robots and robots, and robots and environment ([24, 25, 26, 27, 28])). This passivity is particularly desirable for haptic rendering and interactive simulation, since, as pointed out in [29], it would allow for: 1) stable simulation with *any* parameters and update-

[1] In contrast to quasi-static systems, for which time integration is typically not necessary (e.g., [7, 8, 5])

rate (e.g., arbitrarily light/stiff simulation with coarse integration interval); and 2) *modular* construction of provably-stable virtual world by simply combining individual virtual objects, each designed to be passive. Information delays among virtual objects or virtual worlds can also be addressed in a straightforward manner using this passivity formalism (e.g., see Chap. 5 and [30, 31] for the application of multi-user haptics).

In this regards, we develop the passive midpoint integrator, the time integration derived from the discrete-time passivity principle, for haptic rendering and interactive simulation. The midpoint time integration is well-known for its ability of energy-conserving in the field of structural mechanics (e.g., [21, 23, 22, 32, 33, 34, 35]), yet, to our knowledge, has not been adopted so far for haptic rendering and interactive simulation. This missing of the midpoint integration is particularly surprising for haptic rendering, as its energy-conserving property naturally suggests a possibility of discrete-time (lossless) passive simulation, that was proposed as one of the ultimate goals of haptic rendering in [29, 36], yet, has been dodging so far from any attempts to attain that. Of course, this energy-conserving property is only for a single system, and the passivity property, which concerns with their interconnection with other objects via power-conjugate input-output mapping, has not been studied in the structural mechanics field. It was however independently shown in ([37, 4]) that, by directly enforcing passivity in the discrete-time domain similarly for the derivation of variational integrators (e.g., [38]), it is indeed possible to develop a passive integrator, which is called NPMI (non-iterative passive mechanical integrator) and turns out to be a midpoint integrator. Then, in [39], this NPMI was extended for mechanical systems in maximal coordinates with LCP (linear complementarity problem) formulation of Coulomb friction contact.

In this thesis, we unify and generalize these NPMI-related results of [37, 4, 39] as a midpoint integration framework, that can deal with linear and nonlinear mechanical systems expressed both in generalized and maximal coordinates. Some approximations, inevitable to render the algorithms interactively-fast, are also adopted here as for the most other haptic rendering and interactive simulation results (e.g.,

[13, 11, 12, 10, 14, 6, 15, 22, 23]). Even so, our proposed algorithms can still exhibit fairly passive behaviors, as manifested by our validating simulation examples (see Sec. 3.5) and also as similarly concluded in ([22, 23]) for midpoint integration with some approximations. Due to this reason, we call our integrator **passive midpoint integrator (PMI)**.

We also address the accuracy of the contact clustering for this PMI-based haptic rendering and interactive simulation framework. Contact simulation (among rigid bodies) is one of the key components in any robot simulators. This problem of contact simulation has a long history of research (e.g., [40, 41, 10]) and still remains as an active area of investigation (e.g., [42, 43, 44]), since the contact simulation of many simulators is often not so stable and/or its accuracy not well-established [42]. While deriving both the constraint-based contact solving and the penalty-based contact solving, which effectively enforce discrete-time passivity of the simulation (i.e., stable), we propose a **data-driven/learning-based contact clustering** (i.e., clustering contact points and/or contact normals into a manageable number of clusters). This contact clustering is required, especially when multiple contact are engaged, to achieve reasonably-fast speed of the simulation calculation while also enhancing numerical stability of the contact simulation. However, this contact clustering is less investigated than the contact solving [44, 45, 46, 14, 8, 47] or collision detection (e.g., [48, 49]). In addition, the mapping from the contact clustering to the contact simulation accuracy is, in general, difficult (or, more often, impossible) to derive analytically. So, we employ the machine learning technique for this contact clustering and establish/verify the accuracy of the contact simulation by using real experiment data. For more details of research motivation and formulations, please see chapter 4.

Due to its passivity enforcement and implementation flexibility, our proposed haptic rendering and interactive simulation framework possesses the following advantages: 1) It can stably simulate mechanical objects with wide-range of parameters (e.g., any positive-definite mass, stiffness, and damping gains, and varying update-rate possible), thereby, allowing for easy and modular tuning of virtual objects while also providing capacity to adjust the discrete-time damping dissipation to render the

simulation more or less stable depending on the target performance and behavior; 2) Consequently, it can simulate very light/stiff objects, which is desired for sharper/-transparent haptic rendering or necessary to simulate some classes of mechanical systems yet, typically cannot be stably rendered by (conditionally-stable) SEI-based frameworks (e.g., CHAI-3D); 3) It can simulate marginally-stable/energy-conserving behaviors such as vibration of flexible objects or elastic contact between two rigid objects with no extra restitution models that typically cannot be achieved by dissipative/decaying IEI-based frameworks, with its implementation complexity at least comparable to that of IEI; and 4) It allows us to use both the maximal and generalized coordinates and even their combination, substantially streamlining the simulation of some mechanical systems. To our knowledge, this thesis provides a solution for the first time to the long-standing problem posed in [29, 36], i.e., how to attain passivity and interactivity of simulation at the same time.

1.2 Outline

The outline of this thesis is as follows. In chapter 1, we provides an introduction and statement of the contributions with review of some related works. In chapter 2, we introduce some preliminary materials, including the definition of passivity, the kinematics and dynamics of rigid bodies in SE(3) and robots, and the non-iterative passive mechanical integrator (NPMI) for linear mechanical system in [37] which is the starting point of this thesis derived only for linear mechanical system. We then derive the PMI-based simulation framework both in generalized coordinates and maximal coordinates in chapter 3, while deriving both constraint-based and penalty-based multi-point contact formulation for our PMI-based simulation for each coordinates in Sec. 3.3. We also present the virtual coupling technique in Sec. 3.4 for a passive haptic rendering while proposing the method to improve its transparency. Experimental results of PMI-based simulation, including peg-in-hole task, flexible beam haptic interaction, and UATD hand grasping, are presented in Sec. 3.5. We also propose data-driven learning-based contact clustering framework in chapter 4, for the

accuracy of multi-point contact simulation. The accuracy of this clustering is then present in Sec. 4.4 with its performance comparison with other available robot simulators. Finally, we present the application of the PMI-based simulation for multi-user haptics in chapter 5, which shows that the open-loop passivity of the PMI simulation is crucial for the network delay stabilization. Summary and some comments on the future research directions are followed in chapter 6.

Chapter 2

Preliminary

2.1 Mechanical Systems and Passivity

The virtual worlds, we aim to construct in this thesis, are mainly composed of multiple articulated rigid bodies ([12]) with some constraints (e.g., no penetration, rotation/pivot joints, etc.), elastic elements (e.g., springs), viscous damping elements, and Coulomb friction among them. A part of these mechanical objects is typically interfaced with some human commanding device (e.g., haptic device), the issue of which is not to be discussed here in detail and, instead, we refer readers to the prior results on the (possibly multi-rate) virtual coupling applicable to our PMI simulation (e.g., [25, 50, 4]) and also to Sec. 3.5 for some illustrative examples. Large-scale deformable objects (e.g., human organs [51]) are not the main concern of this thesis either and we leave them and their salient issues (e.g., model reduction, fast contact resolution, large deformation, St. Venant tensor simulation, etc.) for future researsh topic, although the PMI framework is equally applicable for them as well (e.g., [30]).

Then, there are two ways to describe such articulated multiple rigid bodies: 1) *maximal coordinate*, which is the SE(3) rigid body transformation of each rigid body with the relative motion constraints implemented by Lagrange multipliers or some other means (e.g., [12, 52]); and 2) *generalized coordinate*, which is obtained by reducing the degree-of-freedom (DOF) of the total system directly applying their relative motion constraints (e.g., [4, 53]). The maximal coordinate approach is typically pre-

ferred for simulating complex systems (e.g., composed of large number of rigid bodies with possible collisions among them), whereas the generalized coordinate approach is more convenient or imperative for some mechanical systems (e.g., flexible beam with nodal coordinates, rigid links with elastic tendon routing, etc). Our rendering framework proposed in this paper allows for both of these maximal and generalized coordinates as well as their combination (see Sec. 3.5).

More specifically, the maximal coordinate approach simulates the articulated multiple rigid bodies as a combination of their constituent single rigid body, whose dynamics in SE(3) can be written by the following Newton-Euler dynamics s.t.,

$$m\ddot{x} + b_x\dot{x} + d\psi_x^T + \lambda_x = f + f_c \tag{2.1}$$

$$J\dot{\omega} + \omega \times J\omega + b_w\omega + d\psi_w^T + \lambda_w = \tau + \tau_c \tag{2.2}$$

$$\dot{R} = RS(\omega) \tag{2.3}$$

where $x \in \mathrm{E}(3)$ is the position with the mass $m > 0$, $b_x \in \Re^{3\times3}$ is positive semi-definite damping matrix, and $f \in \Re^3$ is to embed the exogenous force (e.g., virtual coupling), all expressed in the inertial frame $\{O\}$; whereas $R \in \mathrm{SO}(3)$ is the rotation matrix of the body frame $\{B\}$ w.r.t. $\{O\}$, $\omega \in \mathrm{so}(3)$ is the angular velocity, $J \in \Re^{3\times3}$ is the (constant) moment of inertia, $b_w \in \Re^{3\times3}$ positive semi-definite damping matrix, τ to embed the exogenous torque, all expressed in body frame $\{B\}$. Here, $d\psi_x, d\psi_w \in \Re^{1\times3}$ are the one-form of certain potential energy $\psi(x, R)$ s.t., $d\psi(x, R)/dt = d\psi_x \cdot \dot{x} + d\psi_w \cdot w$ (e.g., gravity or spring force - see Sec. 3.2.1); $\lambda_x, \lambda_w \in \Re^3$ are to be used to render the constraints among the rigid bodies to realize their articulation, for which we utilize potential-based approach (see Sec. 3.2.2); $f_c, \tau_c \in \Re^3$ are the contact force s.t., $f_c^T \cdot \dot{x} + \tau_c^T \cdot w \leq 0$ (e.g., with Coulomb friction - see Chapter 3.3); and $S(\star)$ is the skew-symmetric operator with $S(a)b = a \times b$, $a, b \in \Re^3$.

On the other hand, the dynamics of articulated rigid bodies in generalized coordinates can be written by the following continuous-time nonlinear Lagrangian dynamics

s.t.,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + B\dot{q} + d\psi^T(q) = f + f_c \tag{2.4}$$

where $q \in \Re^n$ is the configuration (i.e., generalized coordinates), $M(q) \in \Re^{n\times n}$ is the symmetric positive-definite inertia matrix, $C(q,\dot{q}) \in \Re^{n\times n}$ is the Coriolis matrix, $B \in \Re^{n\times n}$ is positive semi-definite damping matrix, $d\psi(q) \in \Re^{1\times n}$ is the one-form of potential energy function $\psi(q) \in \Re$ (e.g., gravity, spring potential) with $d\varphi/dt = d\varphi\cdot\dot{q}$, $f_c \in \Re^n$ is the contact force with $f_c^T\dot{q} \leq 0$ (e.g., with Coulomb friction), and $f \in \Re^n$ is to embed exogenous force (e.g., virtual coupling with haptic device - see Sec. 3.4).

It is then well-known that the mechanical system, represented in maximal coordinates (2.1)-(2.3) or in generalized coordinates (2.4) satisfies the (continuous-time) passivity ([53]):

$$\int_0^T u^T y dt \geq E(T) - E(0), \quad \forall T \geq 0 \tag{2.5}$$

with $E(t) := \kappa(t) + \psi(t)$ being the total energy, where $(u, y, \kappa) = ([f;\tau], [\dot{x};w], (1/2)$ $m\dot{x}^T\dot{x} + (1/2)w^TJw)$ or $(u, y, \kappa) = (f, \dot{q}, (1/2)\dot{q}^TM(q)\dot{q})$ for the maximal coordinate formulation (2.1)-(2.3) or the generalized coordinate formulation (2.4), respectively. This passivity property (2.5) has been the key enabler of numerous nonlinear control techniques for mechanical systems, particularly those exploiting their nonlinear dynamics (2.1)-(2.3) or (2.4) (i.e., passivity-based control [53, 54, 55, 56]) instead of unnecessarily attempting to fully eliminate that (e.g., feedback linearization).

Further, from the passivity theorem [1] that feedback interconnection of *any* passive systems is necessarily stable regardless of their parameters (with some detectability condition met), this passivity property has been arguably the most powerful tool in telerobotics and haptics to ensure human-in-the-loop stability (e.g., [24, 27, 30]). See Fig. 2-1. This also implies that, if we can simulate each virtual entity to be passive, we may then construct very a complex virtual world simply by combining those passive virtual entities, each of which can be designed individually without referring to other components and whose parameters can be set arbitrarily as long as

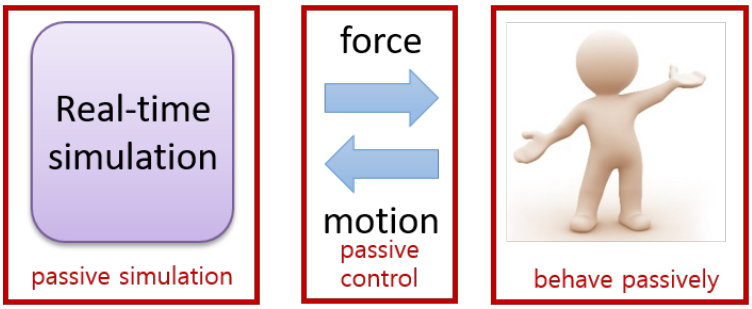


Figure 2-1: Feedback interconnection of passive systems are stable by the passivity theorem [1]. Thus, we can make human-in-the loop simulation stable by passively implementing discrete-time simulation and interfacing between them (i.e., haptic rendering and control).

passivity is preserved (e.g., any positive-definite M, B, K for (2.4), if $M(q) = M$ and $d\psi^T = Kq$) while also theoretically ensuring the stability of the whole virtual world. The benefits of passive simulation, namely, modular design, parameter-invariant stability, and theoretical stability guarantee, were recognized in [29, 36] and suggested as one of the ultimate goals of haptic rendering and interactive simulation.

This (fundamental) continuous-time passivity (2.5), yet, is in general not easily achievable in the discrete-time domain, particularly when the simulation speed needs to be haptically-fast. In fact, the work of [29] shows that there does not exist any explicit discrete-time passive integrators. To address this, in [37], we proposed a new integrator, non-iterative passive mechanical integrator (NPMI), which is implicit, yet, requires no iterations, thus, can still be solved haptically-fast while also enforcing discrete-time passivity. This NPMI turns out to be a midpoint integrator, and is to be generalized in this thesis to be PMI (passive midpoint integrator). In the next Sec. 2.2, we summarize the result of [37], which stands as the backbone of the ensuing development of PMI-based haptic rendering and interactive simulation framework of this thesis.

2.2 Discrete-Time Passivity and NPMI for Linear Mechanical Systems

Passivity is the notion related to not only the storage function but also the input-output power flow. Therefore, it can change whether the system is passive or not depending on the definitions of input, output, and storage function. For the mechanical system, as in (2.5), it is natural to use the total energy for the storage function, force/velocity pair for input/output pair. As for the discrete-time simulation of the mechanical system, the important observations for the development of NPMI [37] are that: 1) how to define the representative velocity and force affects simulation stability, and 2) choosing v_k or v_{k+1} for the counter part of the continuous-time output velocity $v(t)$ cannot fully reflect the energetic behavior of the mechanical system during the simulation time interval $T_k = [t_k, t_{k+1})$ in discrete-time domain.

To introduce the main idea of NPMI [37], let us consider the following simple scalar linear mass-damper-spring dynamics:

$$ma + bv + kx = f \tag{2.6}$$

where $x \in \Re$, $v = \dot{x} \in \Re$, $a = \ddot{x} \in \Re$ are the position, velocity and acceleration, $m, b, k > 0$ are the mass, damper and spring coefficients, and $f \in \Re$ is the exogenous force, respectively. The passivity property (2.5) is then given by: during $T_k = [t_k, t_{k+1})$,

$$\int_{t_k}^{t_{k+1}} fv dt = \frac{1}{2}m(v_{k+1}^2 - v_k^2) + \frac{1}{2}k(x_{k+1}^2 - x_k^2) + \int_{t_k}^{t_{k+1}} bv^2 dt \tag{2.7}$$

where $\star_k := \star(t_k)$. Let us denote by $\hat{\star}_k$ to be "representative" $\star_k$ for the duration T_k. Then, we can approximate the above passivity relation s.t.,

$$\hat{f}_k \hat{v}_k T_k = \frac{1}{2}mv_{k+1}^2 - \frac{1}{2}mv_k^2 + \frac{1}{2}kx_{k+1}^2 - \frac{1}{2}kx_k^2 + b\hat{v}_k^2 T_k \tag{2.8}$$

$$= \frac{1}{2}(v_{k+1} - v_k)m(v_{k+1} + v_k)\frac{1}{2}(x_{k+1} - x_k)k(x_{k+1} + x_k) + b\hat{v}_k^2 T_k \tag{2.9}$$

where we also use $T_k := t_{k+1} - t_k$ with a slight abuse of notation.

Our goal here is then to find a discrete-time integrator map

$$\mathcal{L} : (x_k, v_k, f_k) \to (x_{k+1}, v_{k+1}) \tag{2.10}$$

and the representative power variables $\hat{v}_k, \hat{f}_k$ so that the map $\mathcal{L}$ exactly duplicates the continuous-time passivity relation (2.9), or, equivalently, its differential form

$$f^T v = v(ma + bv + kx). \tag{2.11}$$

By comparing (2.9) with (2.11), the NPMI then suggests the following kinematics relation [37]:

$$\hat{v}_k := \frac{v_{k+1} + v_k}{2} = \frac{x_{k+1} - x_k}{T_k} \tag{2.12}$$

and, further, applying (2.12) to (2.9) and comparing that with (2.11), it also suggests the following dynamics update equation:

$$m\frac{v_{k+1} - v_k}{T_k} + b\hat{v}_k + k\hat{x}_k = f_k \tag{2.13}$$

with $\hat{x}_k := \frac{x_{k+1}+x_k}{2}$ and $\hat{f}_k = f_k$.

This NPMI (2.12)-(2.13) then exactly duplicates the continuous-time passivity property (2.5) by using the "representative" velocity $\hat{v}_k$ and position $\hat{x}_k$. For the choice of $\hat{f}_k$, we have latitude, thus, we choose $\hat{f}_k = f_k$ for simplicity of the integrator $\mathcal{L}$. With these "representative" quantities, the NPMI then satisfies the following discrete-time passivity:

$$\sum_{k=0}^{N-1} f_k \hat{v}_k T_k = \frac{1}{2}m(v_N^2 - v_o^2) + \frac{1}{2}k(x_N^2 - x_o^2) + \sum_{k=0}^{N-1} b\hat{v}_k^2 T_k \tag{2.14}$$

$$\geq -\frac{1}{2}mv_o^2 - \frac{1}{2}kx_o^2 = -E_o \tag{2.15}$$

$\forall N \geq 0$, i.e., the maximum extractable energy from the system is lower bounded by

the system initial energy. Since it is derived directly by enforcing certain physical principle (i.e., passivity) in the discrete-time domain, the NPMI is a geometric integrator similar to variational integrators (e.g., [38]), which discretizes the variational principle itself.

This NPMI (2.12)-(2.13) also turns out to be a midpoint integrator (e.g., [23, 22, 33, 34, 35]), which is well-known for its superior energy-conserving property (i.e., passivity, yet, with no input-output consideration) in structural mechanics community, yet, has not been adopted so far for haptic rendering and interactive simulation. In this thesis, we generalize this NPMI as passive midpoint integrator (PMI), with full consideration of input-output mapping (i.e., power port), generalized and maximal coordinate formalisms, nonlinear dynamics (2.1)-(2.4), and constraints and multi-point contacts with Coulomb friction, all the while retaining its discrete-time passivity and non-iterativeness by utilizing certain coordinate transformation and potential action approximation.

Of course, other types of integrators have also been used for haptic rendering and interactive simulation, namely, explicit Euler integrator (EEI), symplectic Euler integrator (SEI) and implicit Euler integrator (IEI), with the second being adopted for the majority of widely-used open physics engines (e.g., Bullet and ODE, with VREP and CHAI-3D built upon them). In the next Sec. 2.2.1, we perform a simple, yet, effective comparison study among these EEI, SEI, IEI and PMI.

2.2.1 Comparison of EEI, SEI, IEI and PMI

Consider the simple scalar linear mass-spring system, which can be written by

$$m\frac{v_{k+1}-v_k}{T_k}+kx_k=f_k \qquad x_{k+1}=x_k+v_kT_k \tag{2.16}$$

$$m\frac{v_{k+1}-v_k}{T_k}+kx_k=f_k \qquad x_{k+1}=x_k+v_{k+1}T_k \tag{2.17}$$

$$m\frac{v_{k+1}-v_k}{T_k}+kx_{k+1}=f_k \qquad x_{k+1}=x_k+v_{k+1}T_k \tag{2.18}$$

$$m\frac{v_{k+1}-v_k}{T_k}+k\hat{x}_k=f_k \qquad x_{k+1}=x_k+\hat{v}_kT_k \tag{2.19}$$

presented in the order of EEI, SEI, IEI and PMI, where $\hat{x}_k, \hat{v}_k$ are defined in (2.12)-(2.13). Note that the only difference among each integrator is that they adopt different "representative" velocity: v_k (EEI), v_{k+1} (SEI and IEI), and $\hat{v}_k$ (PMI) and different spring force rendering: kx_k (EEI and SEI), kx_{k+1} (IEI), and $k\hat{x}_k$ (PMI).

Then, adopting the procedure similar to that in Sec. 2.2, we can obtain the passivity property of each integrator as follows. First, for the EEI, we have

$$\sum_{k=0}^{N-1} f_k v_k T_k \leq \frac{1}{2}m(v_N^2 - v_0^2) + \frac{1}{2}k(x_N^2 - x_0^2) \tag{2.20}$$

for which we utilize

$$\sum_{k=1}^{N} m\frac{v_{k+1} - v_k}{T_k} v_k T_k \leq \frac{1}{2}mv_{N+1}^2 - \frac{1}{2}mv_0^2 \tag{2.21}$$

$$\sum_{k=1}^{N} kx_k\frac{x_{k+1} - x_k}{T_k} T_k \leq \frac{1}{2}kx_{N+1}^2 - \frac{1}{2}kx_0^2 \tag{2.22}$$

derived by using Cauchy-Schwarz inequality. This then implies that EEI is in general not passive. Similarly, for the IEI, we have

$$\sum_{k=0}^{N-1} f_k v_{k+1} T_k \geq \frac{1}{2}m(v_N^2 - v_0^2) + \frac{1}{2}k(x_N^2 - x_0^2) \tag{2.23}$$

for which we utilize

$$\sum_{k=1}^{N} m\frac{v_{k+1} - v_k}{T_k} v_{k+1} T_k \geq \frac{1}{2}mv_{N+1}^2 - \frac{1}{2}mv_0^2 \tag{2.24}$$

$$\sum_{k=1}^{N} kx_{k+1}\frac{x_{k+1} - x_k}{T_k} T_k \geq \frac{1}{2}kx_{N+1}^2 - \frac{1}{2}kx_0^2. \tag{2.25}$$

This then implies that IEI is passive, yet, in general dissipative, thus, not able to simulate such lossless behaviors as harmonic oscillation. On the other hand, for the SEI, its passivity is undetermined, since its relation is a combination of (2.24) (i.e., passive) with (2.22) (i.e., non-passive). This in fact well matches with the known fact that the SEI is only conditionally stable depending on parameters. Finally, for the

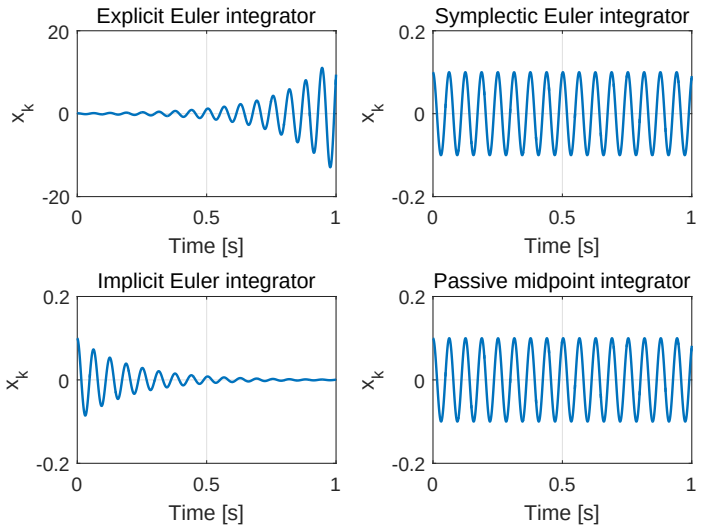


Figure 2-2: Simple harmonic oscillation simulation with EEI, SEI, IEI and PMI: $m = 0.01$[kg], $k = 100$[N/m], $T_k = 0.001$[sec].

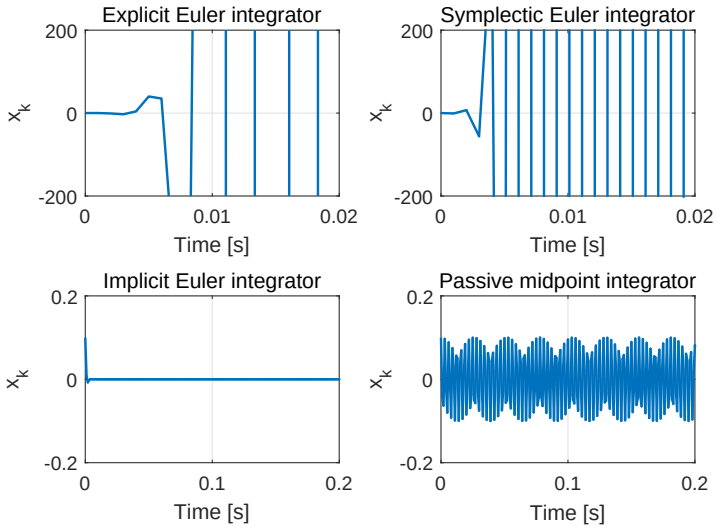


Figure 2-3: Simple harmonic oscillation simulation with EEI, SEI, IEI and PMI: $m = 0.00001$[kg], $k = 100$[N/m], $T_k = 0.001$[sec].

PMI, we have

$$\sum_{k=0}^{N-1} f_k \hat{v}_k T_k = \frac{1}{2} m (v_N^2 - v_0^2) + \frac{1}{2} k (x_N^2 - x_0^2) \tag{2.26}$$

from (2.15), implying that it is passive, and, further, exactly lossless, thus, can precisely simulate harmonic oscillation.

This observation is then confirmed by the simulation results in Fig. 2-2 and Fig. 2-3, where: 1) the EEI always diverges; 2) the SEI can simulate harmonic oscillation with large mass, yet, diverges with small mass; 3) the IEI always converges to zero, yet, not able to sustain harmonic oscillation, clearly showing its dissipativeness; and 4) only the PMI can simulate harmonic oscillation with the total energy exactly conserved for any mass/spring ratio. This then clearly shows the advantage of our PMI, namely, its ability to passive (i.e., stable) simulation with wide-range of parameters as compared to EEI and SEI, while also being typically simpler to implement than IEI on top of its ability to simulate lossless behaviors.

Chapter 3

Development of Passive Midpoint Integrator

3.1 Passive Midpoint Integrator in Generalized Coordinate

The NPMI (2.12)-(2.13) of Sec. 2.2 exactly enforces passivity only for linear mechanical systems. In this section, we first generalize this NPMI to passive midpoint integration (PMI) of nonlnear Lagrange dynamics (2.4) in generalized coordinates. Here, we show that the PMI formulation developed in Sec. 2.2 is not able to enforce passivity for the Lagrange dynamics (2.4) due to the dynamics nonlinearity. To overcome this, we then propose a certain coordinate transformation, with which the nonlinear Lagrange dynamics (2.4) can be converted into the form amicable for the PMI formalism to enforce passivity in the discrete-time domain.

3.1.1 PMI of Nonlinear Lagrange Dynamics

Consider the multi-DOF nonlinear Lagrange dynamics (2.4) with the generalized coordinates $q \in \Re^n$. Here, we focus on quadratic potential functions $\psi(q) = \frac{1}{2}q^T K q$, since the widely used proportional controller of joint space can be obtained from this quadratic potential functions, and we can establish discrete-time passivity exactly

with them without any approximations necessary. Of course, other types of analytic nonlinear potentials $\psi(q)$ can also be used with their potential action approximated as to be presented in Sec. 3.2.2 while preserving the non-iterativeness of the PMI simulation with some damping B injection to enforce the discrete-time passivity even with this approximation.

Now, suppose $M(q)$ in (2.4) be constant (i.e., $M(q) = M$). This means $C(q, \dot{q})$ in (2.4) is skew-symmetric, since $\dot{M} - 2C$ is skew-symmetric. We can then write the PMI expression of (2.4) as a vector version of (2.13), with x_k, v_k replaced by $q_k, v_k := \dot{q}_k \in \Re^n$ and also with $C_k\hat{v}_k := C(q_k, v_k)\hat{v}_k$ added in (2.13). Then, we can show that this PMI of Lagrange dynamics (2.4) with constant M and linear spring potential $d\psi_k^T = Kq_k$ is discrete-time passive, i.e., $\forall N \geq 1$,

$$\sum_{k=0}^{N-1}[f_k + f_{c,k}]^T\hat{v}_k T_k = \sum_{k=0}^{N-1}\left[E_{k+1} - E_k + \hat{v}_k^T C_k \hat{v}_k T_k\right] \tag{3.1}$$

$$= E_{\bar{N}+1} - E_o \geq -E_o \tag{3.2}$$

where C_k is skew-symmetric, $\hat{v}_k := (v_{k+1} + v_k)/2$ is the "representative" velocity during T_k, and $E_k := \frac{1}{2}v_k^T M v_k + \frac{1}{2}q_k^T K q_k$ is the total energy.

This discrete-time passivity (3.2), however, is not granted for non-constant M. That is, from $M(q)$ being configuration-dependent, we have $M_{k+1} := M(q_{k+1}) \neq M_k := M(q_k)$, with which, instead of (3.2), we have:

$$\sum_{k=0}^{N-1}[f_k + f_{c,k}]^T\hat{v}_k T_k = \frac{1}{2}q_N^T K q_N - \frac{1}{2}q_o^T K q_o + \sum_{k=0}^{N-1}\left[\frac{v_{k+1}^T M_k v_{k+1}}{2} - \frac{v_k^T M_k v_k}{2} + \hat{v}_k^T C_k \hat{v}_k T_k\right] \tag{3.3}$$

where, differently from (3.2), the kinetic energies in the bracket do not cancel out with each other over the successive T_k and T_{k+1}, and C_k is not skew-symmetric either. This is because the skew-symmetricity of $\dot{M} - 2C$ (or, equivalently, $\dot{M} = C + C^T$) cannot be simulated during T_k, since doing so requires future information M_{k+1}, C_{k+1} at the onset of T_k. To address this, here, we utilize the coordinate transformation proposed

in ([4]), with which the nonlinear Lagrange dynamics (2.4) is transformed into another Lagrangian form with constant inertia and skew-symmetric Coriolis term, thus, can still be passively integrated by using PMI similar to (2.13).

More precisely, define the following (continuous-time) coordinate transformation:

$$\xi := M^{\frac{1}{2}}(q)\dot{q}, \quad \delta := M^{-\frac{1}{2}}(q)f, \quad \delta_c := M^{-\frac{1}{2}}(q)f_c \tag{3.4}$$

with which the kinetic energy can be written as

$$\frac{1}{2}\dot{q}^T M(q)\dot{q} = \frac{1}{2}\xi^T I \xi \tag{3.5}$$

where $I \in \Re^{n\times n}$ is the identity matrix, implying that, for the transformed velocity ξ, the inertia will be this constant I. Using

$$I\dot{\xi} = M^{\frac{1}{2}}(q)\ddot{q} + \tfrac{d}{dt}[M^{\frac{1}{2}}(q)]\dot{q} \tag{3.6}$$

we can then rewrite (2.4) s.t.

$$I\dot{\xi} + Q(q,\dot{q})\xi + M^{-\frac{1}{2}}(q)Kq = \delta + \delta_c \tag{3.7}$$

where the Coriolis-like term $Q(q,\dot{q}) \in \Re^{n\times n}$ is defined by

$$Q(q,\dot{q}) := M^{-\frac{1}{2}}CM^{-\frac{1}{2}} - \tfrac{d}{dt}[M^{\frac{1}{2}}]M^{-\frac{1}{2}} \tag{3.8}$$

with arguments omitted for brevity.

This matrix Q in (3.8) is in fact skew-symmetric. That is, with M being symmetric, we have

$$Q + Q^T = M^{-\frac{1}{2}}[C + C^T]M^{-\frac{1}{2}} - \tfrac{d}{dt}[M^{\frac{1}{2}}]M^{-\frac{1}{2}} - M^{-\frac{1}{2}}\tfrac{d}{dt}[M^{\frac{1}{2}}] \tag{3.9}$$

$$= M^{-\frac{1}{2}}\tfrac{d}{dt}[M^{\frac{1}{2}}M^{\frac{1}{2}}]M^{-\frac{1}{2}} - \tfrac{d}{dt}[M^{\frac{1}{2}}]M^{-\frac{1}{2}} - M^{-\frac{1}{2}}\tfrac{d}{dt}[M^{\frac{1}{2}}] = 0 \tag{3.10}$$

where we use the fact that $\dot{M} = C + C^T$ from $\dot{M} - 2C$ being skew-symmetric. We

can also show that the transformed dynamics (3.7) preserves the same passivity as (2.5): $\forall T \geq 0$,

$$\int_0^T [f_c + f]^T \dot{q} dt = \int_0^T [\delta_c + \delta]^T \xi dt \tag{3.11}$$

$$= \int_0^T \left[\xi^T I \dot{\xi} + \xi^T Q \xi + \xi^T M^{-\frac{1}{2}} K q \right] dt \tag{3.12}$$

$$= E(T) - E(0) \geq -E(0) \tag{3.13}$$

where $E := \frac{1}{2}\dot{q}^T M(q)\dot{q} + \frac{1}{2} q^T K q = \frac{1}{2}\xi^T I \xi + \frac{1}{2} q^T K q$ is the total energy.

Since the transformed dynamics (3.7) has constant inertia I and skew-symmetric Q, we can now passively simulate (3.7) similar to (2.13) s.t.

$$I \frac{\xi_{k+1} - \xi_k}{T_k} + Q_k \frac{\xi_{k+1} + \xi_k}{2} + M_k^{-\frac{1}{2}} K \frac{q_{k+1} + q_k}{2} = \delta_k + \delta_{c,k} \tag{3.14}$$

where $\delta_k := M_k^{-\frac{1}{2}} f_k$, $\delta_{c,k} := M_k^{-\frac{1}{2}} f_{c,k}$, and $Q_k := Q(q_k, v_k)$. We can then show that

$$\sum_{k=1}^{N-1} [\delta_k + \delta_{c,k}]^T \hat{\xi}_k T_k = \sum_{k=0}^{N-1} \left[\kappa_{k+1} - \kappa_k + \hat{\xi}_k^T Q_k \hat{\xi}_k T_k + \hat{\xi}_k^T M_k^{-\frac{1}{2}} K \frac{q_{k+1} + q_k}{2} T_k \right] \tag{3.15}$$

$$= \kappa_N - \kappa_o + \sum_{k=0}^{N-1} \hat{\xi}_k^T M_k^{-\frac{1}{2}} K \frac{q_{k+1} + q_k}{2} T_k \tag{3.16}$$

where $\hat{\xi}_k := (\xi_{k+1} + \xi_k)/2$, $\kappa_k := \xi_k^T I \xi_k / 2 = v_k^T M_k v_k / 2$, and we use the fact that Q_k is skew-symmetric.

This (3.16) then also suggests the following kinematics law, similar to (2.12), yet, transformed via $M_k^{\frac{1}{2}}$, s.t.

$$\frac{q_{k+1} - q_k}{T_k} = M_k^{-\frac{1}{2}} \frac{\xi_{k+1} + \xi_k}{2} \tag{3.17}$$

with which makes the last summation term of (3.16) be $\frac{1}{2} q_N^T K q_N - \frac{1}{2} q_o^T K q_o$. Indeed, this is consistent with the continuous-time transformation (3.4).

Note that, to obtain the passive discrete-time simulation of the continuous-time Lagrangian dynamics (2.4), we first transform it to (3.7); then discretize it via (3.14);

and connect to the configuration q of the original dynamics through (3.17). One of the important part is to compute skew-symmetric Q_k in (3.14). To compute Q_k via (3.8), we may also use the following (discrete-time) Sylvester equation [57] to obtain $\dot{M}_k^{\frac{1}{2}}$, that is, given $M_k^{\frac{1}{2}},\ C_k$, find $\dot{M}_k^{\frac{1}{2}}$ s.t.,

$$\dot{M}_k^{\frac{1}{2}} M_k^{\frac{1}{2}} + M_k^{\frac{1}{2}} \dot{M}_k^{\frac{1}{2}} = \dot{M}_k = [C_k + C_k^T] \tag{3.18}$$

with which we can also show that $Q_k = M_k^{-\frac{1}{2}} C_k M_k^{-\frac{1}{2}} - \dot{M}_k^{\frac{1}{2}} M_k^{-\frac{1}{2}}$ is skew-symmetric. More precisely, this matrix equation can be rewritten as a $n^2 \times n^2$ linear system,

$$(I_n \otimes M_k^{\frac{1}{2}} + M_k^{\frac{1}{2}} \otimes I_n)\text{vec}(\dot{M}_k^{\frac{1}{2}}) = \text{vec}(C_k + C_k^T) \tag{3.19}$$

where $I_n \in \Re^{n \times n}$ is the identity matrix, $\otimes$ is the Kronecker product, and vec is the vectorization operator. Note also that since M_k is positive-definite and symmetric $M_k^{\frac{1}{2}}$ is also positive-definite, symmetric, and unique. Therefore, the matrix term $(I_n \otimes M_k^{\frac{1}{2}} + M_k^{\frac{1}{2}} \otimes I_n)$ of (3.19) also symmetric and positive definite. This can be solved quickly by using the Cholesky decomposition. On the other hand, we can further reduce the size of the linear system (3.19) by using the symmetric property. Again, since M_k is symmetric, $\dot{M}_k^{\frac{1}{2}}$ should also symmetric. This means that there are $\frac{n(n-1)}{2}$ number of same terms in $\text{vec}(\dot{M}_k^{\frac{1}{2}})$ and $\text{vec}(C_k + C_k^T)$. Thus, by rearranging (3.19), we can reduce its size from n^2 to $n(n+1)/2$. During the experiment of Sec. 4.4 where 7-DOF robot arm simulated, solving (3.19) without size reducing takes 0.8$\sim$0.9[ms]. If the reducing is applied it takes 0.7$\sim$0.8[ms].

In the next Sec. 3.1.2, we analyse the symplecticity of this PMI simulation (3.14) and (3.17). Symplecticity is one of the important property of the Hamiltonian dynamics which especially affects to the long term dynamic behavior [9, 58]. One of the main reason why the symplectic Euler integration (SEI) is widely used for haptic rendering and interactive simulation, even though it is conditionally stable, is that it preserves this symplecticity. Therefore, we analyse the symplecticity of our PMI simulation and compare/discuss overall performance and advantages of our PMI over the SEI.

3.1.2 Comparison of PMI with Symplectic Integrators

To define symplectic integrators for the Lagrangian dynamics (2.4), let us first define its Hamiltonian s.t.,

$$H(p,q) = p^T M(q)^{-1} p + U(q) \tag{3.20}$$

where $q \in \Re^n$ is the configuration, $p := M(q)\dot{q} \in \Re^n$ is the conjugate momenta, and $U(q) \in \Re$ is the potential energy. Then, the Hamiltonian dynamics formulation of (2.4) is given by

$$\dot{p} = -\frac{\partial H}{\partial q}(p,q), \qquad \dot{q} = \frac{\partial H}{\partial p}(p,q) \tag{3.21}$$

which can be written in the following matrix form:

$$\dot{y} := \begin{pmatrix} \dot{p} \\ \dot{q} \end{pmatrix} = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix}^{-1} \begin{pmatrix} \frac{\partial H}{\partial p} \\ \frac{\partial H}{\partial q} \end{pmatrix} =: \mathsf{J}^{-1} \cdot \nabla H(y) \tag{3.22}$$

where $y := [p; q] \in \Re^{2n}$ is the coordinate in the tangent bundle, $\nabla H(y) = [\frac{\partial H}{\partial p}; \frac{\partial H}{\partial q}] \in \Re^{2n}$ is the gradient of H, and $\mathsf{J} \in \Re^{2n \times 2n}$ is the (skew-symmetric) canonical structure matrix. The solution of the Hamiltonian system (3.22) can then be written by its flow map ψ_t with $\psi_t(y_0) := y(t, y_0)$.

Three fundamental properties the Hamiltonian system (3.22) possesses, namely, energy conservation (or passivity), momentum conservation, and symplecticity. Since the first two are familiar, we explain here only the symplecticity. A linear mapping $A : \Re^{2n} \to \Re^{2n}$ is called *symplectic* if

$$w(A\zeta_1, A\zeta_2) = w(\zeta_1, \zeta_2) \tag{3.23}$$

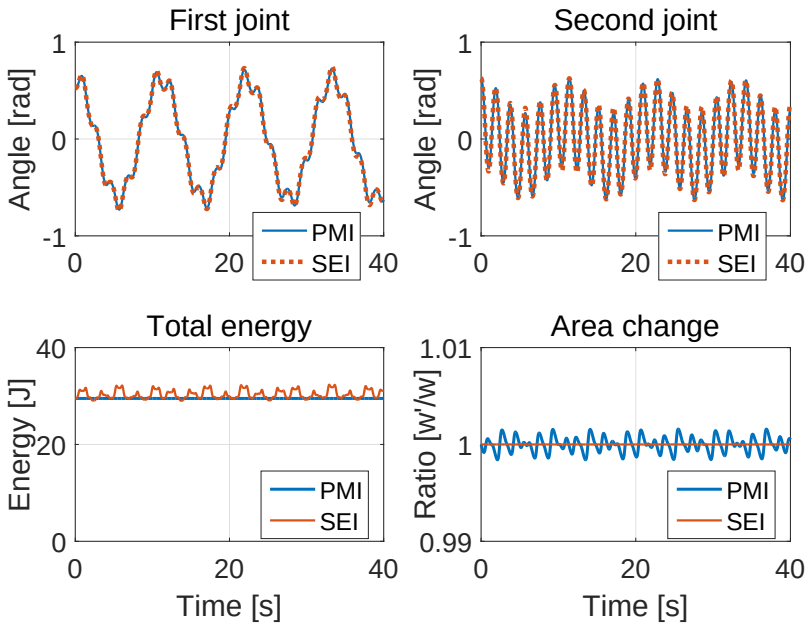


Figure 3-1: Simulation of two serial-link robot arm with linear joint springs with SEI and PMI: the SEI is symplectic, yet, not exactly energy-preserving, whereas the PMI is energy-preserving, yet, not exactly symplectic.

for $\zeta_1 := [\zeta_1^p, \zeta_1^q], \zeta_2 := [\zeta_2^p, \zeta_2^q] \in \Re^{2n}$ with $\zeta_1^p, \zeta_1^q, \zeta_2^p, \zeta_2^q \in \Re^n$, where

$$w(\zeta_1, \zeta_2) := \sum_{i=1}^{n} \det \begin{pmatrix} \zeta_{1,i}^p & \zeta_{2,i}^p \\ \zeta_{1,i}^q & \zeta_{2,i}^q \end{pmatrix} = \sum_{i=1}^{n} (\zeta_{1,i}^p \zeta_{2,i}^q - \zeta_{1,i}^q \zeta_{2,i}^p) \tag{3.24}$$

which is the sum of the oriented areas, i.e., symplecticity implies the area preservation through the map A. For the Hamiltonian system (3.22), the symplecticity condition is given by

$$\left(\frac{\partial \psi_t}{\partial y_0}\right)^T \mathsf{J} \left(\frac{\partial \psi_t}{\partial y_0}\right) = \mathsf{J}. \tag{3.25}$$

Symplectic integrators are those that preserve this simplecticity property of the continuous-time Hamiltonian system (3.22). The following symplectic Euler integra-

tion (SEI) and implicit midpoint integration (IMI) are those symplectic integrators

$$y_{k+1} = y_k + \mathsf{J}^{-1} \cdot \nabla H(p_{k+1}, q_k)T \tag{3.26}$$

$$y_{k+1} = y_k + \mathsf{J}^{-1} \cdot \nabla H(p_k, q_{k+1})T \tag{3.27}$$

$$y_{k+1} = y_k + \mathsf{J}^{-1} \cdot \nabla H(\hat{p}_k, \hat{q}_k)T \tag{3.28}$$

with the first two being the SEI and the last the IMI, where $y_k := [p_k; q_k] \in \Re^{2n}$, $T > 0$ is the integration step, and $\hat{\star}_k := \frac{\star_{k+1}+\star_k}{2}$. Both of these SEI and IMI are known to be symplectic and momentum conserving, yet, not energy preserving, where the symplecticity condition is given by

$$\left(\frac{\partial y_{k+1}}{\partial y_k}\right)^T \mathsf{J} \left(\frac{\partial y_{k+1}}{\partial y_k}\right) = \mathsf{J}. \tag{3.29}$$

Now, let us see if the PMI (3.14)-(3.17) is also symplectic. For this, from (3.14)-(3.17), we can obtain the following equation:

$$\begin{aligned}\begin{pmatrix} p_{k+1} \\ q_{k+1} \end{pmatrix} = \begin{bmatrix} M_{k+1}^{\frac{1}{2}}(q_{k+1}) & 0 \\ 0 & I \end{bmatrix} \left(\frac{I}{T_k} + S_k(p_k, q_k)\right)^{-1} \\ \times \left(\frac{I}{T_k} - S_k(p_k, q_k)\right) \begin{bmatrix} M_k^{-\frac{1}{2}}(q_k) & 0 \\ 0 & I \end{bmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix}\end{aligned} \tag{3.30}$$

where $p_k = M^{\frac{1}{2}}(q_k)\xi_k \in \Re^n$ and $S_k(p_k, q_k) := \frac{1}{2}\left[Q_k(p_k, q_k), M_k^{-\frac{1}{2}}(q_k)K;\ M_k^{-\frac{1}{2}}(q_k), 0\right] \in \Re^{2n\times 2n}$.

To derive (3.29) from (3.30) turns out too complicated to analytically perform. Note however that we have $M_{k+1}^{\frac{1}{2}}$ in (3.30), which, in general, can take any positive definite matrix, suggesting the PMI would be unlikely symplectic. For this, here, we resort to a numerical approach, that is, measure how the area of a certain shape in the space of $(p, q) \in \Re^{2n}$ evolves in the course of the PMI integration of (3.14)-(3.17). The results are shown in Fig. 3-1, where we can see that the PMI is energy-preserving (from passivity), yet, not symplectic, although it still stays near to be symplectic. We can show that the PMI is not momentum-preserving either. This then means that

the PMI is not the IMI, although, for linear mechanical systems, they become the same.

Even so, we believe the PMI has advantages for haptic rendering and interactive simulation as compared to those symplectic integrators particularly due to the following two powerful properties: 1) non-iterativeness and 2) passivity. Note first from (3.27)-(3.28) that the symplectic integrators should in general solve the nonlinear map (i.e., ∇H) real-time in an iterative manner due to the presence of the same variables both in the left-hand side and inside of $\nabla H(\star)$ in (3.27)-(3.28). This iterative nature of the symplectic integrators, combined with their non-passivity (i.e., instability possible), poses a challenging issue in practice, that is, this nonlinear map must be solved precisely to maintain stability, which may be time-consuming for complex ∇H or requires only small time-step T to ensure fast-enough convergence of this nonlinear map solving, both not so suitable for real-time interactivity of simulation. From its being dissipative, the IEI, in contrast, is known to be less sensitive to this solution precision and often chosen over the SEI or IMI when this issue becomes important.

On the other hand, note from (3.14)-(3.17) that the PMI is non-iterative with no necessity to solve such a nonlinear map. This non-iterativeness of the PMI is attained due to the coordinate transformation (3.4) (for nonlinearity from inertia) and the potential action approximation (e.g., (3.54) for potential nonlinearity). Of course, with this approximate action of nonlinear potentials, the passivity will be also in general only approximate. This "leakage" of passivity, yet, can be rather easily resolved by injecting (typically small) linear damping B, which can be still simulated quickly due to the non-iterativeness of the PMI and whose passivity also still guaranteed due to the passivity of the PMI. This unconditionally stable/passive damping allows for the so called "controllable damping" (e.g., increase (or decrease, resp.) B for more stable (or less stable, resp.) behaviors while preserving simulation stability), which is not possible for those symplectic integrators, theoretically or practically (i.e., precision of nonlinear map solving). These non-iterativeness and passivity, we believe, are more relevant than symplecticity and momentum-preserving for haptic rendering and interactive simulation, since the stability and real-time interactivity are their most

fundamental requirements and their simulated objects are also anyway typically under the influence of frequent human/contact forces differently from those in celestial mechanics or molecular simulation, where long-term isolated behavior simulation is more important.

3.2 Passive Midpoint Integrator in Maximal Coordinate

In this section, we generalize the NPMI (2.12)-(2.13) of Sec. 2.2 to passive midpoint integration (PMI) of rigid bodies in SE(3), which constitute the basic building block of the maximal coordinate formalism of articulated rigid bodies. We also show how to incorporate constraints into this PMI-based simulation of rigid bodies to realize articulation among the rigid bodies. The main complication for this is that the configuration space of rigid bodies in maximal coordinate is SE(3), which is not a vector space. To circumvent these challenges, here, we utilize some approximations in enforcing passivity. Such approximation is in general necessary for other haptic rendering or interactive simulation results to attain interactivity (e.g., contact force linearization with IEI or SEI [14]) or for structural dynamics to speed up the computation (e.g., interaction explicitation [35, 22, 23]), which yet is known to work adequately in most cases as also confirmed through our experiments (see Sec. 3.5).

3.2.1 PMI of Rigid Body in SE(3)

Consider the Newton-Euler dynamics (2.1)-(2.3) of rigid bodies in SE(3). To derive the PMI formulation of (2.1)-(2.3), here we only consider the inertial dynamic (i.e., associated with m, J), the potential action (i.e., $d\psi_\star$), and the exogenous forcing f, τ. The constraint and contact forces (i.e., $\lambda_\star, f_c, \tau_c$) will be added in Sec. 3.2.2 and Chap. 3.3, respectively. We also restrict the potential ψ in this Sec. 3.2.1 to be the linear E(3) spring directly acting on x (i.e., $\psi = \psi_x = \frac{1}{2}x^T k_x x$) or the SO(3) spring acting on R (i.e., $\psi = \psi_w(R)$), while postponing the treatment of more general $\psi(x, R)$ to

Sec. 3.2.2 and Chap. 3.3.

Then, with the linear spring potential $\psi_x = \frac{1}{2}x^T k_x x$, the translation dynamics (2.1) can simply be simulated via the linear NPMI (2.12)-(2.13) while exactly enforcing passivity as stated in Sec. 2.2. The attitude dynamics (2.2) also has the structure similar to that of linear NPMI (2.13) except the coupling term $w \times Jw$. From the same motivation of Sec. 3.1.1 that constant inertia and skew-symmetric Coriolis matrices enable us to directly adopt (2.13) for the attitude dynamics (2.2) which is expressed in its body frame $\{\mathrm{B}\}$, we can rewrite the coupling term $w \times Jw$ as $w \times Jw = -S(Jw)w$, where $S(Jw) \in \Re^{3\times 3}$ is skew-symmetric with $a^T S(Jw)a = 0$ $\forall a \in \Re^3$. This then suggests the following PMI expression of the attitude dynamics (2.2):

$$J\frac{\omega_{k+1} - \omega_k}{T_k} - S(J\omega_k)\hat{\omega}_k + b_w\hat{\omega}_k + d\psi_{w_k}^T = \tau_k \tag{3.31}$$

with the "representative" angular velocity

$$\hat{w}_k := \frac{w_{k+1} + w_k}{2} \tag{3.32}$$

similar to (2.12). It is then not difficult to show the following passivity (lossless) property with $\psi_w = 0$ similar to (2.15):

$$\sum_{k=0}^{N-1} \tau_k^T \hat{\omega}_k T_k = \frac{1}{2}w_N^T J w_N - \frac{1}{2}w_o^T J w_o \geq -\frac{1}{2}w_o^T J w_o \tag{3.33}$$

where we use $\hat{w}_k^T S(Jw_k)\hat{w}_k = 0$, $\forall k = 0, ..., N-1$. Note that this coupling term $S(Jw)w$ is often omitted in computer graphics literatures. For this case, the PMI expression (3.31) is simply reduced to the linear NPMI (2.13).

One thing still missing here is how to connect the angular velocity $\hat{w}_k \in \Re^3$ in (3.32) to update the rotation matrix $R_k \in \mathrm{SO}(3)$ while ensuring passivity as much as permissible for interactivity requirement. For this, we utilize the following discrete-

time version of the rotational kinematics (2.3) s.t.,

$$R_{k+1} = R_k \exp(S(\hat{\omega}_k T_k)) \tag{3.34}$$

where $\exp(\star)$ is the exponential map [59]. This discrete-time kinematics equation (3.34) in SO(3) then should be compatible with the SO(3) spring potential, as so are true for $\hat{v}_k, \hat{x}_k$ and $\psi_k = \frac{1}{2}kx_k^2$ in Sec. 2.2. For this, define the following SO(3) spring potential:

$$\psi_w := \frac{1}{2}k_w\phi^2(R, R_o) \tag{3.35}$$

with $k_w > 0$ and

$$\phi(R, R_o) := \| \log(R^T R_o) \| =: \cos^{-1}\gamma \tag{3.36}$$

where $R_o \in \mathrm{SO}(3)$ is the reference rotation, $\mathrm{tr}(\star)$ and $\log(\star)$ are the trace and matrix logarithm operators, and

$$\gamma = \frac{1}{2}[\mathrm{tr}(R^T R_o) - 1] \tag{3.37}$$

This ϕ is known to be the geodesic on the unit sphere [60]. This SO(3) spring, of course, is practically important as well, e.g., directly applicable to implement virtual coupling between two large-rotation rigid bodies (e.g., virtual coupling for palm and fingers of multi-fingered hand - see Fig. 3-3).

Our objective is then to find the action $d\psi_{w_k}$ for (3.31) to implement the SO(3) spring (3.35) under the passivity requirement. For this, particularly due to that SO(3) is not a vector space differently from that for the linear NPMI (2.13), we attempt to enforce passivity only approximately, which, yet, turns out to work fairly well as shown below. More precisely, we aim to define $d\psi_{w_k}$ s.t.,

$$d\psi_{w_k}\hat{\omega}_k T_k = \frac{\partial \psi_w}{\partial \phi}\Delta\phi_k + \frac{1}{2}\frac{\partial^2 \psi_w}{\partial \phi^2}\Delta\phi_k^2 \tag{3.38}$$

which is the second-order approximation of exact potential difference (i.e., $\psi_{w,k+1} - \psi_{w_k}$), with $\partial\psi_w/\partial\phi = k_w\phi_k$ and $\partial^2\psi_w/\partial\phi^2 = k_w$. For (3.38), we also use

$$\Delta\phi_k = \frac{\partial\phi}{\partial\gamma}\Delta\gamma_k, \quad \Delta\gamma_k := \frac{1}{2}\hat{\omega}_k^T\bar{\omega}_k T_k \tag{3.39}$$

where $\partial\phi/\partial\gamma = -1/\sin\phi_k$ and $\bar{\omega}_k \in \Re^3$ is defined s.t.,

$$S(\bar{\omega}_k) := R_k^T R_o - R_o^T R_k \tag{3.40}$$

which has the direction of the equivalent axis between R_k and R_o and the magnitude $\|\bar{\omega}_k\| = 2\sin(\phi_k)$; and the term $\Delta\gamma_k$ is designed s.t.,

$$\Delta\gamma_k = \frac{1}{2}\,\mathrm{tr}[R_{k+1}^T R_o - R_k^T R_o] \approx \frac{1}{2}\,\mathrm{tr}[S^T(\hat{\omega}_k T_k)R_k^T R_o] = \frac{1}{4}\,\mathrm{tr}[S^T(\hat{\omega}_k T_k)(R_k^T R_o - R_o^T R_k)] \tag{3.41}$$

which is reduced to (3.39), where we use (3.34), the first order approximation of $\exp(S(\hat{w}_k T_k)) \approx I + S(\hat{w}_k T_k)$, and the properties of $\mathrm{tr}(\star)$, e.g., $\mathrm{tr}(A+B) = \mathrm{tr}(A) + \mathrm{tr}(B)$, $\mathrm{tr}(ABC) = \mathrm{tr}(BCA)$, $\mathrm{tr}(A) = \mathrm{tr}(A^T)$, and $a^T b = \mathrm{tr}[S^T(a)S(b)/2]$, $\forall a, b \in \Re^3$.

Applying these to (3.38), we can then define $d\psi_{w_k}$, s.t.,

$$d\psi_{w_k}^T = -\frac{k_w\phi_k}{2\sin\phi_k}\bar{\omega}_k + \frac{1}{8}\frac{k_w T_k}{\sin^2\phi_k}(\bar{\omega}_k\bar{\omega}_k^T)\hat{\omega}_k \tag{3.42}$$

where $\bar{\omega}_k\bar{\omega}_k^T \in \Re^{3\times3}$ is a rank one matrix. Since $||\bar{\omega}_k|| = 2\sin(\phi_k)$, we can further approximately simplify $d\psi_{w_k}$ s.t.,

$$d\psi_{w_k}^T \approx -\frac{k_w\phi_k}{2\sin\phi_k}\bar{\omega}_k + \frac{k_w T_k}{2}\hat{\omega}_k \tag{3.43}$$

where note that the second term acts similarly as viscous damping action. The passivity of this SO(3) spring action (3.43) is then demonstrated in Fig. 3-2, where stable sustained oscillation is attained with the total energy almost precisely preserved even if the action (3.43) is designed to enforce passivity only approximately.

Finally, the PMI expression of the Newton-Euler dynamics (2.1)-(2.2) with the

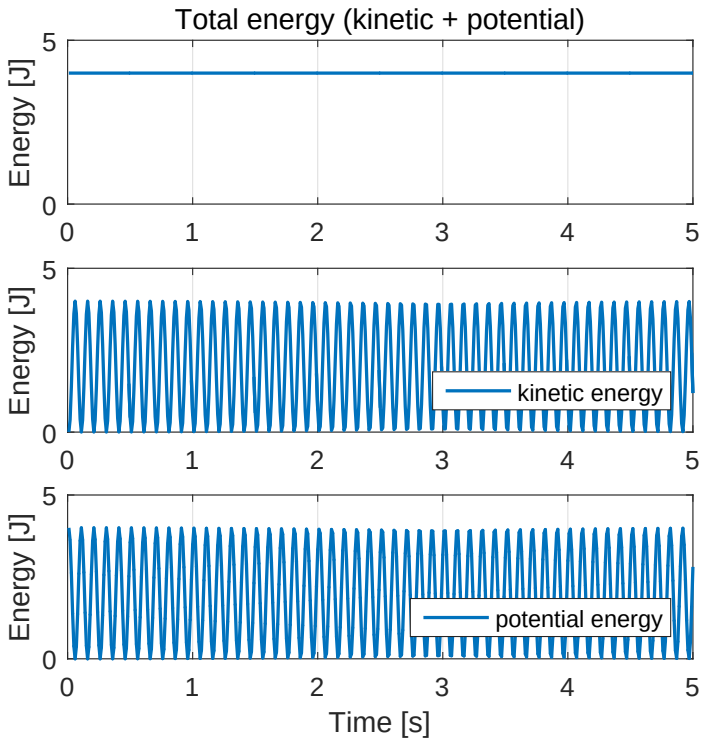


Figure 3-2: Total energy of rotational dynamics (3.31) with SO(3) spring (3.43) with nearly perfect energy conservation and stable/sustained oscillation in SO(3): with $J = 0.001 \cdot I[\text{kg m}^2]$, $k_w = 1$ [Nm/rad] and $T_k = 0.01$ [sec] starting from $\phi_o = 0.9\pi$ along the direction of $\frac{1}{\sqrt{6}}(1, 2, 1)$.

linear spring for x and the SO(3) spring potential action (3.43) can be written in one single matrix equation:

$$M\frac{V_{k+1} - V_k}{T_k} + C_k(\omega_k)\hat{V}_k + B\hat{V}_k + d\psi_k^T = F_k \tag{3.44}$$

where $V_k := [v_k; \omega_k] \in \Re^6$ with $\hat{V}_k := [\hat{v}_k; \hat{\omega}_k] \in \Re^6$, $M := \text{diag}[mI_{3\times3}, J] \in \Re^{6\times6}$ is the inertia matrix, $C_k := \text{diag}[0_{3\times3}, -S(J\omega_k)]$ with $C_k = -C_k^T$, $B := \text{diag}[b_x, b_\omega] \in \Re^{6\times6}$ is the damping matrix, $d\psi_k^T := [k_x\hat{x}_k; d\psi_{\omega_k}^T] \in \Re^6$ is the potential action as defined in (2.13) and (3.43), and $F_k := [f_k; \tau_k] \in \Re^6$ is the exogenous wrench. This equation can also be written in the following form, which is to be used for the ensuing developments:

$$\hat{M}_k\hat{V}_k = M_kV_k - d\bar{\psi}_k^T + F_k \tag{3.45}$$

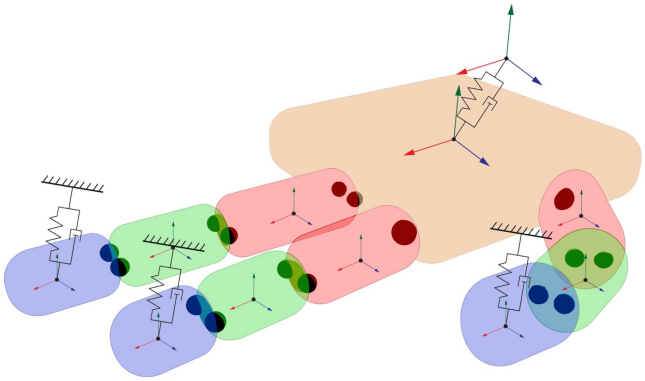


Figure 3-3: Multi-fingered hand rendered as articulated multiple rigid bodies with joints implemented by point constraint springs.

where $\hat{M}_k := [M_k + C_k + B + (T_k/2)K] \in \Re^{6\times 6}$ with $M_k := 2M/T_k$, $K := \text{diag}[k_x, k_\omega] \in \Re^{6\times 6}$ is the spring matrix from (2.13) and (3.43), and $d\bar{\psi}_k^T := [-k_x x_k; \frac{k_\omega \phi_k}{2\sin\phi_k}\bar{\omega}_k] \in \Re^6$ from (3.43). Here, both $\hat{M}_k$ and M_k are non-singular.

3.2.2 PMI-Based Rendering of Articulated Rigid Bodies

One of the most widely used entities in haptic rendering and interactive simulation are so called articulated rigid bodies (e.g., [12]). Here, we propose to construct such articulated rigid bodies as a collection of multiple rigid links with their joints implemented by constraining position of some points on the successive links via very stiff spring. For instance, see the multi-fingered hand in Fig. 3-3, where the distal interphalangeal (DIP) and metacarpophalangeal (MCP) joints of the index finger are implemented by enforcing two-point position constraints on distal-middle and middle-proximal phalanxes; whereas the thumb carpometacarpal (CMC) joint by one single-point position constraint between the palm and the metacarpal bone. Other types of joint constraints can also be implemented in a similar manner by using different spring potentials (e.g., elliptic joint with four-point springs [61]; prismatic joint with a SO(3)-potential and a two-directional translation spring potential, etc.).

Consider m rigid-links with l-point position constraints (e.g., $m = 10$ and $l = 16$ for the hand in Fig. 3-3). We assume each of these l-points be shared only by two successive links, although the development presented here can easily be extended to

more general cases. Then, we can define the following map

$$c_1, c_2 : \{1, 2, ..., l\} \mapsto \{1, 2, ..., m\} \tag{3.46}$$

i.e., connecting the i-th position constraining point ($i \in \{1, 2, ..., l\}$) to its presence on the $c_1(i)$-th and the $c_2(i)$-th links. We can then denote this presence of the i-th constraining-point by

$$p_{i,j} := x_{c_j(i)} + R_{c_j(i)} r_{c_j(i),i} \in \Re^3, \quad j = 1, 2 \tag{3.47}$$

which is on the $c_j(i)$-th link with the offset from $x_{c_j(i)}$ given by $r_{c_j(i),i} \in \Re^3$ expressed in the $c_j(i)$-th link body fixed frame.

To implement the joints among the m-link, we then want to enforce

$$\Delta p_i := p_{i,1} - p_{i,2} \equiv 0 \tag{3.48}$$

Stacking up all the l-point position constraints, define the constraint function $G : \Re^{3l} \to \Re^{3l}$ s.t.,

$$G := [\Delta p_1; \Delta p_2; ...\Delta p_l] \tag{3.49}$$

with $G = 0$ implying that all the l-point position constraints are ensured, and so are the joints among the m-links. We can then first-order approximate this G s.t.,

$$G_{k+1} \approx G_k + J_k \hat{V}_k T_k \tag{3.50}$$

where $\hat{V}_k = [\hat{v}_{1_k}; \hat{\omega}_{1_k}; \hat{v}_{2_k}; \hat{\omega}_{2_k}; ...; \hat{v}_{m_k}; \hat{\omega}_{m_k}] \in \Re^{6m}$ contains the representative velocities of all the m rigid bodies, and $J \in \Re^{3l \times 6m}$ is the Jacobian defined s.t., $dG/dt = JV$.

The PMI expression of the dynamics of the m rigid links with the l-constraints

can then be written similar to (3.45) s.t.,

$$\hat{M}_k\hat{V}_k = M_kV_k - d\bar{\psi}_k^T + J_k^T\lambda_k + F_k \tag{3.51}$$

where $\lambda_k \in \Re^{3l}$ is to embed certain potential action to robustly maintain the constraint $G = 0$ similar to the penalty-based contact formulation [47, 62], and all the other terms are defined for the m rigid bodies similarly to those in (3.45). Our goal is then to design the potential action λ_k to attain the constraint $G = 0$ while also respecting the passivity requirement. For this, let us define the following potential function associated with the (target) constraint $G = 0$:

$$\psi_{\lambda_k} := \frac{1}{2}k_\lambda G_k^T G_k \tag{3.52}$$

where $k_\lambda > 0$ is a constant.

Then, using (2.15), (3.33), (3.38), (3.50) and (3.51), we can obtain

$$F_k^T\hat{V}_kT_k \approx \kappa_{k+1} - \kappa_k + \psi_{k+1} - \psi_k + \hat{V}_k^TB\hat{V}_kT_k - \hat{V}_k^TJ_k^T\lambda_kT_k \tag{3.53}$$

where $\kappa_k := \frac{1}{2}V_k^TMV_k$ is the kinetic energy, M is the inertia matrix and $\psi_k = \frac{1}{2}x_k^Tk_xx_k + \psi_{w,k}$ of the m rigid bodies collective as defined before (3.45). Further, while utilizing the approximated G_{k+1} (3.50) and

$$\psi_{\lambda_{k+1}} - \psi_{\lambda_k} \approx k_\lambda G_k^TJ_k\hat{V}_kT_k + \frac{1}{2}k_\lambda\hat{V}_k^TJ_k^TJ_k\hat{V}_kT_k^2 \tag{3.54}$$

we can design

$$\lambda_k = -k_\lambda[G_k + \frac{1}{2}J_k\hat{V}_kT_k] \tag{3.55}$$

with which the above relation (3.53) reduces to: with $B = 0$,

$$F_k^T\hat{V}_kT_k \approx E_{k+1} - E_k \tag{3.56}$$

where $E_k := \frac{1}{2}V_k^T M V_k + \psi_k + \psi_{\lambda,k}$ is the total energy. With this λ_k of (3.55), we can rewrite (3.51) as

$$\left[\hat{M}_k + \frac{k_\lambda T_k}{2} J_k^T J_k\right] \hat{V}_k = M_k V_k - k_\lambda J_k^T G_k - d\bar{\psi}_k^T + F_k \tag{3.57}$$

which implies that $\hat{V}_k$ can be obtained without iteration even with the implicit constraining force λ_k (3.55).

The analysis above shows that the potential action λ_k (3.55) would robustly enforce the constraint $G = 0$ while also nearly enforcing the passivity of the PMI simulation of articulated rigid bodies. Here, note that the passivity is enforced only approximately, as we adopt the potential action approximation (3.54), which in turn stems from our adoption of the Jacobian approximation (3.50). We have this nonlinear potential approximation (3.54) even if the potential ψ_{λ_k} itself is only quadratic, since the maximal coordinates are on a non-vector space SE(3). This potential action approximation (3.54) is crucial to render the PMI simulation still non-iterative even with the nonlinear potential ψ_{λ_k}. The "lack" of passivity due to this potential action approximation (3.54) can also be easily amended by injecting (typically) small amount of damping B, which can also be incorporated into the PMI simulation while retaining its non-iterative structure (3.57).

This passive constraint-enforcing $G = 0$ turns out to be very useful, since it allows us to utilize very large stiffness (i.e., k_λ) to enforce the joint articulations, even for very light rigid links (e.g., to render light virtual proxy for transparency) - see Sec. 3.5.1. Note that we can also implement the spring potential action $\psi(x, R)$ (e.g., spring connected not on the center-of-mass x of rigid bodies) by using the same procedure to design λ_k proposed here, that is, define the spring connection point positions similar to G, define their spring potentials similar to ψ_λ, and design the potential action $d\psi$ similar to λ_k to respect the approximated potential difference equation. Here, we also adopt this "soft" potential action ψ_{λ_k} to enforce the constraint $G = 0$ instead of the "hard" Lagrange multiplier approach for (3.50), since: 1) the latter is much less robust (e.g., drift) with the Jacobian approximation (3.50) than the former, which contains

potential action similar to the penalty-based contact formulation [47, 62]; and 2) the passivity of the former is easily achievable (with small damping B injection), whereas that of the latter still unclear.

3.2.3 Improving Real-Time Interactivity

Note that, during implementation of (3.57), we should calculate the inversion matrix of $\left[\hat{M}_k + \frac{k_\lambda T_k}{2} J_k^T J_k\right]$ for every simulation steps. For this, the term $\hat{M}_k$ is the form of a block-diagonal matrix while the term $J_k^T J_k$ has off-diagonal terms. This requires $\Re^{6m\times 6m}$ matrix inversion which takes so many computation time (e.g., for 10 rigid bodies in Fig. 3-3, about 2 3 ms for a matrix inversion of $\Re^{60\times 60}$). To reduce this computation time and improve the real-time interactivity, here we introduce a modelling which is slightly modified from the original ones, yet can be solved faster without loss of the passivity.

Main ideas is to used step delayed term for $\frac{k_\lambda T_k}{2} J_k^T J_k$ in (3.57) of Sec. 3.2.2. Physically, this term means a joint damping matrix between connected rigid bodies. To simplify and get some insight, let us assume that there are only two rigid bodies. Then, the damping force can be rewritten as can be written

$$F_{D,k} := \frac{k_\lambda T_k}{2} J_k^T J_k \hat{V}_k = -\frac{k_\lambda T_k}{2} \begin{bmatrix} J_{1,k} & -J_{2,k} \end{bmatrix}^T \begin{bmatrix} J_{1,k} & -J_{2,k} \end{bmatrix} \hat{V}_k \tag{3.58}$$

$$= \frac{k_\lambda T_k}{2} \begin{bmatrix} J_{1,k}^T J_{1,k} & -J_{1,k}^T J_{2,k} \\ -J_{2,k}^T J_{1,k} & J_{2,k}^T J_{2,k} \end{bmatrix} \begin{bmatrix} \hat{V}_{1,k} \\ \hat{V}_{2,k} \end{bmatrix} \tag{3.59}$$

where $J_{i,k} \in \Re^{6\times 6}$ with $i = [1, 2]$ is a Jacobian from the center of mass of each rigid body to the joint position $p_{i,j}$ of (3.47). Here, the problematic part is the off-diagonal terms. The off-diagonal terms are non-zero and 6 by 6 matrix (i.e., $J_{i,k}^T J_{j,k} \in \Re^{6\times 6}$). So, when it is integrated with $\hat{M}_k$ in (3.57) where $\hat{M}_k$ has only 6 by 6 matrices for its diagonal part, it makes $\left[\hat{M}_k + \frac{k_\lambda T_k}{2} J_k^T J_k\right]$ being full matrix, which requires many computation time to calculate its inversion. If we can eliminate the off-diagonal terms, then we can calculate the inversion by calculating several inversions of 6 by 6 diagonal terms. In this regards, we redesign the following partially step-delayed damping for

$F_{D,k}$, s.t.,

$$F_{D,k} \approx \frac{k_\lambda T_k}{2} \left(\begin{bmatrix} J_{1,k}^T J_{1,k} & 0 \\ 0 & J_{2,k}^T J_{2,k} \end{bmatrix} \begin{bmatrix} \hat{V}_{1,k} \\ \hat{V}_{2,k} \end{bmatrix} - \begin{bmatrix} 0 & J_{1,k}^T J_{2,k-1} \\ J_{2,k}^T J_{1,k-1} & 0 \end{bmatrix} \begin{bmatrix} \hat{V}_{1,k-1} \\ \hat{V}_{2,k-1} \end{bmatrix} \right) \tag{3.60}$$

where $J_{i,k-1}\hat{V}_{i,k-1}$ implies the constraint space velocity at previous integration step t_{k-1}. Note that, since the right term of (3.60) is in explicit form, we can eliminate off-diagonal terms and solve (3.57) with only 6 by 6 matrix inversions.

Now, let us check whether this step-delayed damping force is passive or not. For this, the energetics of the original coupling term (3.59) is then given by

$$S_{D,k} = \hat{V}_k^T F_{D,k} = \begin{bmatrix} \hat{V}_{1,k} \\ \hat{V}_{2,k} \end{bmatrix}^T \left(\frac{k_\lambda T_k}{2} \begin{bmatrix} J_{1,k}^T J_{1,k} & -J_{1,k}^T J_{2,k} \\ -J_{2,k}^T J_{1,k} & J_{2,k}^T J_{2,k} \end{bmatrix} \begin{bmatrix} \hat{V}_{1,k} \\ \hat{V}_{2,k} \end{bmatrix} \right) \tag{3.61}$$

$$= \frac{k_\lambda T_k}{2} \left(\|J_{1,k}\hat{V}_{1,k}\|^2 + \|J_{2,k}\hat{V}_{2,k}\|^2 - 2\hat{V}_{1,k}^T J_{1,k}^T J_{2,k} \hat{V}_{2,k} \right). \tag{3.62}$$

This then has a range of

$$0 \leq S_{D,k} \leq k_\lambda T_k \left(\|J_{1,k}\hat{V}_{1,k}\|^2 + \|J_{2,k}\hat{V}_{2,k}\|^2 \right) \tag{3.63}$$

where we use the Cauchy–Schwarz inequality for the inequality. The right terms of (3.63) means that the maximum energy dissipation of the joint constraining damping force (3.59). On the other hand, the energetics of (3.60) is now given as

$$S_{D,k} = \frac{k_\lambda T_k}{2} (\|J_{1,k}\hat{V}_{1,k}\|^2 + \|J_{2,k}\hat{V}_{2,k}\|^2 - \hat{V}_{1,k}^T J_{1,k}^T J_{2,k-1} \hat{V}_{2,k-1} - \hat{V}_{2,k}^T J_{2,k}^T J_{1,k-1} \hat{V}_{1,k-1}). \tag{3.64}$$

Again, with the Cauchy–Schwarz inequality, this then has the following range

$$0 \leq S_{D,k} \leq \sum_{i=1,2} \frac{k_\lambda T_k}{2} \left(\frac{3}{2} \|J_{i,k}\hat{V}_{i,k}\| + \frac{1}{2} \|J_{i,k-1}\hat{V}_{i,k-1}\| \right). \tag{3.65}$$

This means that, in the worst case, the energy dissipation of (3.60) is zero while, in the best case, it is equivalent to $k_\lambda T_k \left(\|J_{1,k}\hat{V}_{1,k}\| + \|J_{2,k}\hat{V}_{2,k}\| \right)$ when we considering back and forth dissipation terms.

Consequently, while considering the best case of (3.63) and the worst case of (3.65), we can solve the inversion of $\left[\hat{M}_k + \frac{k_\lambda T_k}{2} J_k^T J_k\right]$ fast without loss of the passivity by using local damping which amounts to $k_\lambda T_k \, \mathrm{diag}[J_{1,k}^T J_{1,k}, J_{2,k}^T J_{2,k}]\hat{V}_k$. In the real experiment, it takes about 1.5 ms for simulation of four hands in Fig. 3-3 with (3.60) while that of (3.59) takes about 8$\sim$9 ms.

3.3 PMI-Based Multi-Point Contact Simulation

In this section, we introduce how to implement a multi-point contact within the PMI-based rigid body simulation. This contact problem has been one of the key problems in computer graphics, structural mechanics, and haptic rendering. For this, there are two main directions of research, namely, constraint-based method (e.g., [13, 63]) and penalty-based method (e.g., [64, 47]). The penalty-based method generates repulsive elastic force when two objects overlap with each other. This method is simple and easy to implement, yet, it has its inherent limitation, i.e., allowing violation of the contact constraint (i.e., penetration). It is in general not straightforward either to implement Coulomb friction force with this penalty-based method. On the other hand, the constraint-based method directly calculates contact force, that is necessary to satisfy the contact constraint. By doing so, this method prevents the violation of the constraint (to the extent of collision detection error and numerical precision error though, which can induce constraint drift). Friction generation is also more convenient with the constraint-based method though it requires more computation than the penalty-based method In the following sections, we first introduce how to implement constraint-based contact method within the PMI simulation of the previous Sec. 3.1-Sec. 3.2. After then, we introduce the penalty-based contact method while remarking the advantage of the PMI simulation for formulating the penalty-based method.

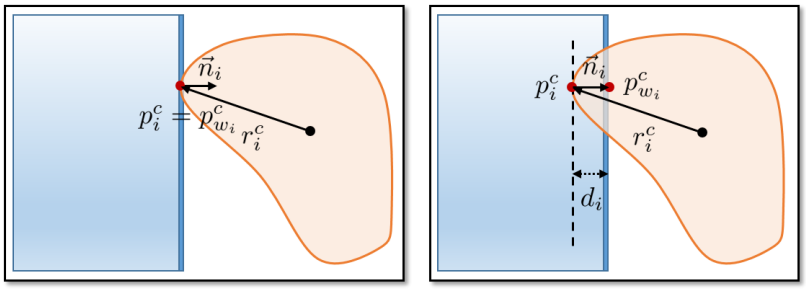


Figure 3-4: Contact configuration of single rigid body contact: $\vec{n}_i$ is the normal direction, p_i^c the contact point on the object boundary, $p_{w_i}^c$ the contact point on the wall boundary, and δ_i is the penetration. Constraint-based method (left) enforces no penetration constraint while penalty-based method (right) allows some penetration.

3.3.1 Constraint-Based Contact Solving

Here, we first formulate the constraint-based contact force for the PMI-based rigid body simulation of Sec. 3.2 and then extend it for the PMI-based Lagrange dynamics simulation of Sec. 3.2. To render the contact among rigid objects, we adopt the point contact model with Coulomb friction. Consider a contact point between two colliding rigid bodies. We can then define the contact normal direction $\vec{n}(t) \in \Re^3$ and the penetration depth along that direction. (See Fig. 3-4 for its illustration.) Denote this depth by $d(t) \geq 0$, with $d(t) = 0$ and $d(t) > 0$ respectively implying contact and no contact. Whenever $d(t) = 0$ occurs, the following higher-order conditions can be utilized to enforce no penetration constraint:

$$\dot{d} \geq 0, \quad \ddot{d} \geq 0, \quad \cdots \tag{3.66}$$

with the non-negative signs allowing only for separation or keeping the contact. Here, we use the velocity-level constraint to avoid the Painlevé paradox, which can then be written as the following complementarity condition [65]:

$$0 \leq \dot{d} \perp \lambda^n \geq 0 \tag{3.67}$$

where $\lambda^n \geq 0$ is the contact force along $\vec{n}$.

Now, suppose that the articulated rigid bodies (3.51) or (3.57) interact with an-

other virtual rigid object, and denote the i-th contact point from that contact by

$$p_i^c := x_{c(i)} + r_i^c \tag{3.68}$$

which is on the $c(i)$-th link and occurs at the offset of $r_i^c \in \Re^3$ from its center-of-mass (expressed in the inertial frame), where $c : \{1, 2, ..., l_c\} \to \{1, 2, ..., m\}$ is the mapping from the contact points to their associated rigid link from the articulated rigid bodies (3.57). The contact wrench, which is to be embedded in F_k of (3.57) is then given by

$$F_{i,k}^c = \underbrace{\begin{bmatrix} I & -S(r_{i,k}^c)R_{\sigma(i),k} \end{bmatrix}}_{=:Ad_{i,k}^c \in \Re^{3\times 6}}{}^T f_{i,k}^c \in \Re^6 \tag{3.69}$$

where $Ad_{i,k}^c$ is the (partial) contact adjoint operator [59] and $f_{i,k}^c \in \Re^3$ is the point contact force with Coulomb friction. See Fig. 3-4 for the illustration of this contact configuration. Then, the contact force $f_{i,k}^c \in \Re^3$ is defined following the point contact and Coulomb friction model [65] s.t.,

$$f_{i,k}^c = \vec{n}_k \lambda_{i,k}^n + \left[\vec{t}_{i,k}^1, \vec{t}_{i,k}^2, \cdots, \vec{t}_{i,k}^{\bar{p}}\right] \left[\lambda_{i,k}^{t_1}; \lambda_{i,k}^{t_2}; \cdots; \lambda_{i,k}^{t_{\bar{p}}}\right] \tag{3.70}$$

$$=: N_{i,k}\lambda_{i,k}^n + D_{i,k}\lambda_{i,k}^t \tag{3.71}$$

where $\lambda_{i,k}^n \geq 0$ is the normal contact force, $\lambda_{i,k}^t \in \Re^{\bar{p}}$ is the tangential friction forces, for which we utilize the $\bar{p}$-sides pyramidal cone approximation [13], $\vec{n}_{i,k}, \vec{t}_{i,k}^j \in \Re^3$ are the normal and tangential direction vectors, and $N_{i,k} \in \Re^3, D_{i,k} \in \Re^{3\times\bar{p}}$ are the stacking matrix of $\vec{n}_{i,k}$ and $\vec{t}_{i,k}^j$.

The PMI expression of the articulated rigid bodies, the virtual object, and their contact interaction can then be obtained by augmenting (3.57) s.t.,

$$\underbrace{\begin{bmatrix} \hat{M}_k + \frac{k_\lambda T_k}{2} J_k^T J_k & 0 \\ 0 & \hat{M}_k^o \end{bmatrix}}_{=:A_k \in \Re^{6(m+1)\times 6(m+1)}} \underbrace{\begin{pmatrix} \hat{V}_k \\ \hat{V}_k^o \end{pmatrix}}_{=:\hat{V}_k^a \in \Re^{6(m+1)}} = \underbrace{\begin{pmatrix} M_k V_k - k_\lambda J_k^T G_k - d\bar{\psi}_k^T \\ M_k^o V_k^o \end{pmatrix}}_{=:B_k \in \Re^{6(m+1)}} + F_k^c \tag{3.72}$$

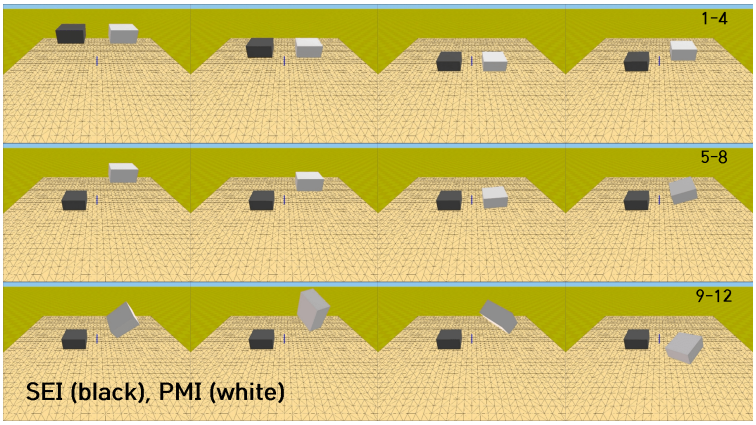


Figure 3-5: Snapshots of rigid box falling on rigid level surface: it bounces back and forth under PMI-LCP (white box), whereas completely plastically collides under SEI (black box) even with the same parameter setting.

with the contact force term $F_k^c \in \Re^{6(m+1)}$ defined by

$$F_k^c := \bar{N}_k \lambda_k^n + \bar{D}_k \lambda_k^t \tag{3.73}$$

where $\lambda_k^n := [\lambda_{1,k}^n; \lambda_{2,k}^n; ...; \lambda_{l_c,k}^n] \in \Re^{l_c}$ and $\lambda_k^t := [\lambda_{1,k}^t; \lambda_{2,k}^t; ...; \lambda_{l_c,k}^t] \in \Re^{l_c \times \bar{p}}$ are the collection of the normal contact and tangential friction forces of all the l_c contact points, and the matrices $\bar{N}_k \in \Re^{6(m+1)\times l_c}$ and $\bar{D}_k \in \Re^{6(m+1)\times(l_c \times \bar{p})}$ are the matrices specifying the directions of λ_k^n and λ_k^t with $Ad_{i,k}^c$ in (3.69) embedded in them. We can then rewrite the above PMI simulation equation s.t.,

$$\hat{V}_k^a = A_k^{-1}(B_k + \bar{N}_k \lambda_k^n + \bar{D}_k \lambda_k^t) \tag{3.74}$$

Similar to (3.67), here, we adopt the velocity-level constraint. We can then write the following complementarity conditions [66]:

$$0 \leq \sigma_k^n \perp \lambda_k^n \geq 0 \tag{3.75}$$

$$0 \leq \sigma_k^t \perp \lambda_k^t \geq 0 \tag{3.76}$$

$$0 \leq \sigma_k^f \perp \lambda_k^f \geq 0 \tag{3.77}$$

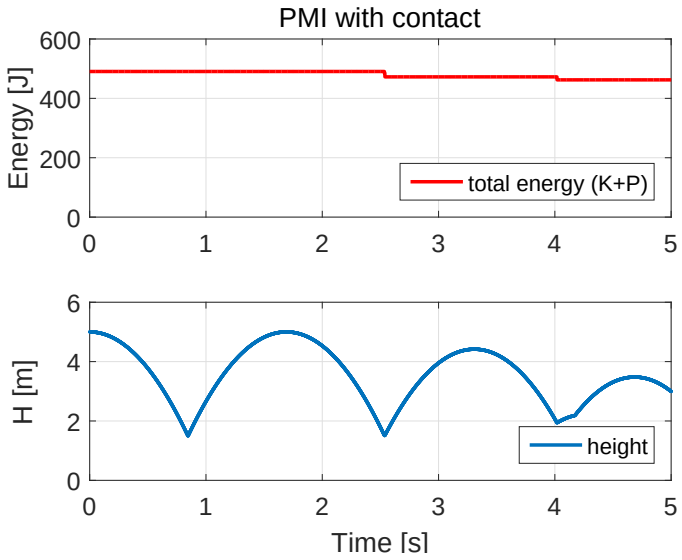


Figure 3-6: Profiles of total energy and height with PMI-LCP (3.79)-(3.80): elastic-normal and dissipative-tangential slips contact attained with total energy remained constant during flight, whereas strictly decreased during contact. Sudden changes of H around 4-4.2[sec] are due to collision with flipping.

where

$$\sigma_k^n := \bar{N}_k^T \hat{V}_k^a \in \Re^{l_c}, \quad \sigma_k^t := \bar{D}_k^T \hat{V}_k^a + U\lambda_k^f \in \Re^{l_c \times \bar{p}}, \quad \sigma_k^f := \bar{\mu}\lambda_k^n - U^T \lambda_k^t \in \Re^{l_c} \tag{3.78}$$

with $\bar{\mu} := \mu I_{l_c \times l_c}$ and $U := \text{diag}[1_{\bar{p}}, 1_{\bar{p}}, ...1_{\bar{p}}] \in \Re^{(l_c \times \bar{p}) \times \bar{p}}$. Here, (3.75) and (3.76) respectively specify the complementarity conditions along the normal and tangential directions, whereas (3.77) the Coulomb friction model with the friction cannot exceed $\mu\lambda^n$. Injecting (3.74) into (3.75)-(3.77), we can then attain the following PMI-LCP (linear complementarity problem) contact formulation:

$$\begin{pmatrix} \sigma_k^n \\ \sigma_k^t \\ \sigma_k^f \end{pmatrix} = \begin{bmatrix} \bar{N}_k^T A_k^{-1} \bar{N}_k & \bar{N}_k^T A_k^{-1} \bar{D}_k & 0 \\ \bar{D}_k^T A_k^{-1} \bar{N}_k & \bar{D}_k^T A_k^{-1} \bar{D}_k & U \\ \mu I_{l_c \times l_c} & -U^T & 0 \end{bmatrix} \begin{pmatrix} \lambda_k^n \\ \lambda_k^t \\ \lambda_k^f \end{pmatrix} + \begin{pmatrix} \gamma_k^n \\ \gamma_k^t \\ 0 \end{pmatrix} \tag{3.79}$$

$$\sigma_k^n, \sigma_k^t, \sigma_k^f, \lambda_k^n, \lambda_k^t, \lambda_k^f \geq 0, \quad \sigma_k^{nT} \lambda_k^n = \sigma_k^{tT} \lambda_k^t = \sigma_k^{fT} \lambda_k^f = 0 \tag{3.80}$$

where $\gamma_k^n := \bar{N}_k^T A_k^{-1} B_k$ and $\gamma_k^t = \bar{D}_k^T A_k^{-1} B_k$. This contact PMI-LCP (3.79)-(3.80) can be solved by using the interior point method or Lemke pivoting algorithm [67].

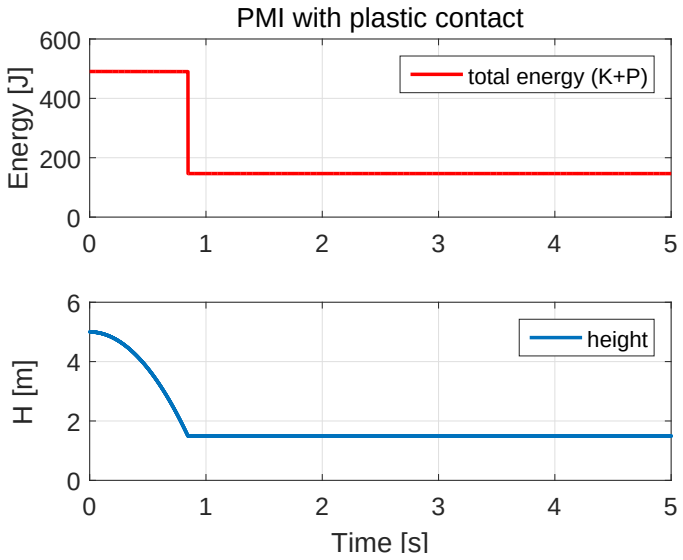


Figure 3-7: Profiles of total energy and height under PMI-LCP (3.79)-(3.80) with $\hat{V}_k^a$ replaced by V_{k+1} for (3.75)-(3.76): completely plastic collision attained as in Fig. 3-8 under SEI.

The complementarity conditions (3.75)-(3.77) naturally suggest passivity of our contact PMI-LCP formulation (3.79)-(3.80), i.e., lossless if no tangential slip occurs or dissipative if it occurs. This passivity of PMI-LCP (3.79)-(3.80) is clearly exhibited in Fig. 3-6, where a rigid box is free falling on a rigid level surface and bounces back and forth from that, with the total energy being constant during the flight stages, while strictly decreasing during the contact stages, during which dissipative tangential slips occur. Note that this simulates lossless (i.e., elastic) collision along the normal direction, whereas dissipative slips along the tangential directions, with no deformation allowed between the two rigid objects. With no slips, this would result in completely lossless/elastic collision.

This ability of our PMI-LCP (3.79)-(3.80) to naturally render (marginally-stable) elastic contact is in a stark contrast to other contact rendering results based on IEI or SEI, which typically can only produce plastic contact unless some extra velocity restitution models are deployed. This is in fact due to the adoption of v_{k+1} as the representative velocity by the IEI and SEI (see Sec. 2.2.1). See Fig. 3-8, where the same box-falling scenario of Fig. 3-6 results in purely plastic collision with the SEI. Of course, in many applications, plastic collisions are necessary, which can also be

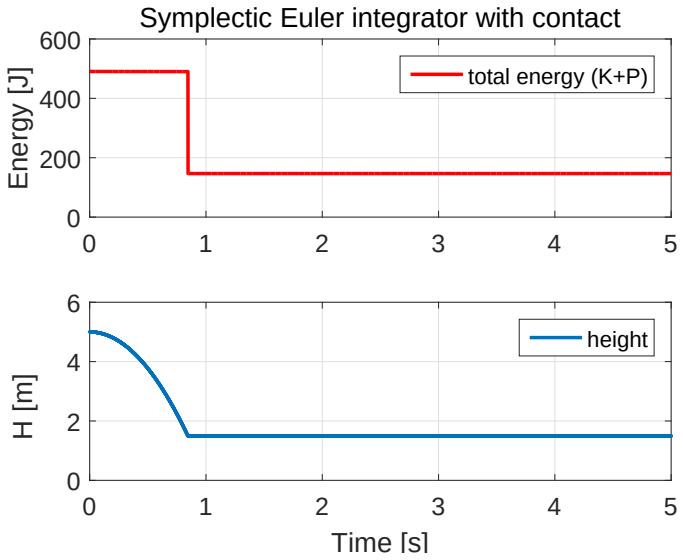


Figure 3-8: Profiles of total energy and height with SEI: although the setting is the same as Fig. 3-6, completely plastic collision is attained (see also Fig. 3-5).

easily achieved by our PMI-LCP (3.79)-(3.80) simply replacing $\hat{V}_k^a$ in σ_k^n, σ_k^t by V_{k+1} - see Fig. 3-7.

Here, note that, in contrast to that of Sec. 3.2.2 (i.e., approximate passivity of λ_k (3.55)), this passivity of the PMI-LCP is exact, since we do not adopt such a Jacobian approximation as done in (3.50). Consequently, the damping injection, added to enforce passivity of the approximate potential action in Sec. 3.2.2, is not necessary here either. Our PMI-LCP algorithm is formulated at the velocity-level (i.e., (3.67)), thus, suffers from the issue of constraint drift (induced by initial collision detection error or numerical precision error). This constraint drift can be easily remedied by incorporating some drift-correction term into γ_k^n of (3.79)-(3.80) as similarly done in [6].

Now, following the above procedure, we can utilize the point contact model with Coulomb friction for the multi-DOF Lagrange system (3.14) interacting with a static virtual rigid object. Here, we choose this static object for simplicity: the derivation presented below can be easily applied to multiple moving rigid objects, which can then be rendered by using the PMI expression in maximal coordinates of Sec. 3.2. We also adopt the velocity level complementary condition as in (3.67) and the $\bar{p}$-

pyramidal cone approximation as in (3.71), and denote by l_c the number of contact points between the Lagrange system (3.14) and the object.

Then, the generalize contact force with the coordinate transformation (3.4), which is to be embedded in δ_k at the i-th contact point ($i \in \{1, 2, ..., l_c\}$) can be written by

$$\delta_{i,k}^c = M_k^{-\frac{1}{2}} f_{i,k}^c = M_k^{-\frac{1}{2}} J_{c_i,k}^T f_{i,k}^c \in \Re^n \tag{3.81}$$

where $J_{c_i,k}(q_k) \in \Re^{3\times n}$ is the Jacobian of the i-th contact point, and $f_{i,k}^c$ is the point contact force with Coulomb friction (3.71).

Combining (3.81) for all the l_c contact points, we can then obtain the total contact force δ_{c_k} in (3.14) s.t.,

$$\delta_{c_k} = \bar{N}_k \lambda_k^n + \bar{D}_k \lambda_k^t \tag{3.82}$$

which is similar to (3.73), yet slightly different notation of $\bar{N}_k$ and $\bar{D}_k$

$$\bar{N}_k := M_k^{-\frac{1}{2}} \left[J_{c_1,k}^T N_{1,k}, J_{c_2,k}^T N_{2,k}, ... J_{c_{l_c},k}^T N_{l_c,k} \right] \in \Re^{n\times l_c} \tag{3.83}$$

$$\bar{D}_k := M_k^{-\frac{1}{2}} \left[J_{c_1,k}^T D_{1,k}, J_{c_2,k}^T D_{2,k}, ... J_{c_{l_c},k}^T D_{l_c,k} \right] \in \Re^{n\times (l_c \times \bar{p})} \tag{3.84}$$

with which we can also rewrite the PMI expression of the Lagrange dynamics (3.14) similar to (3.74), i.e.,

$$\hat{\xi}_k = A_k^{-1} \left[B_k + \bar{N}_k \lambda_k^n + \bar{D}_k \lambda_k^t \right] \tag{3.85}$$

where $A_k := \frac{2}{T_k} I + Q_k + \frac{1}{2} M_k^{-\frac{1}{2}} K M_k^{-\frac{1}{2}} T_k$ and $B_k := \frac{2}{T_k} \xi_k - M_k^{-\frac{1}{2}} K q_k$.

Given this $\hat{\xi}_k$ in (3.85), we can compute the velocity of all the l_c contact points s.t.,

$$\hat{V}_k^c = J_{c_k} M_k^{-\frac{1}{2}} \hat{\xi}_k \in \Re^{3l_c} \tag{3.86}$$

where $J_{c_k} := \left[J_{c_1,k}; J_{c_2,k}; ... J_{c_{l_c},k} \right] \in \Re^{3l_c \times n}$ is the combined Jacobian of the l_c contact

points. The contact complementarity conditions [66] can be written similar to (3.75)-(3.77) s.t.,

$$0 \leq \sigma_k^n \perp \lambda_k^n \geq 0, \quad 0 \leq \sigma_k^t \perp \lambda_k^t \geq 0, \quad 0 \leq \sigma_k^f \perp \lambda_k^f \geq 0 \tag{3.87}$$

where

$$\sigma_k^n := N_k^T \hat{V}_k^c = \bar{N}_k^T \hat{\xi}_k \in \Re^{l_c} \tag{3.88}$$

$$\sigma_k^t := D_k^T \hat{V}_k^c + U\lambda_k^f = \bar{D}_k^T \hat{\xi}_k + U\lambda_k^f \in \Re^{l_c \times \bar{p}} \tag{3.89}$$

$$\sigma_k^f := \bar{\mu}\lambda_k^n - U^T \lambda_k^t \in \Re^{l_c} \tag{3.90}$$

where $N_k := \text{diag}[N_{1,k}, N_{2,k}, ...N_{l_c,k}] \in \Re^{3l_c \times l_c}$, $D_k := \text{diag}[D_{1,k}, D_{2,k}, ...D_{l_c,k}] \in \Re^{3l_c \times (l_c \times \bar{p})}$, and $\bar{\mu} := \mu I_{l_c \times l_c}$ and $U := \text{diag}[1_{\bar{p}}, 1_{\bar{p}}, ..1_{\bar{p}}] \in \Re^{(l_c \times \bar{p}) \times \bar{p}}$ as for (3.75)-(3.77). Finally, we can construct the PMI-LCP formulation in generalized coordinate which is exactly same form of (3.79)-(3.80) with (3.82) - (3.85).

3.3.2 Penalty-Based Contact Solving

Though the PMI-LCP in Sec. 3.3.1 can emulate constraint enforcing contact force, it can suffer from high computation cost for solving (3.79)-(3.80). It is known that even only normal contact can be solved in polynomial time or exponential time in worst-case. Therefore, to be implemented in real-time, numerical method such as clustering should be adopted to reduce the number of contact points (see Chap. 4 for more details of clustering). In this context, in this section, we provide penalty-based contact force formulation which can be solved fast, even though it allows some penetration. The major difference between penalty-based method and constraint-based method is in computing $\lambda_{i,k}^n$ in (3.71). While the constraint-based method mathematically finds it to maintain the non-penetrating constraint, in penalty-based method, it is simply calculated by using elastic force with its penetration $d_{i,k}$. See Fig. 3-4 (right).

Now, to formulate penalty-based contact force with the PMI-based rigid body

simulation, let us consider the following quadratic potential for i-th contact

$$\psi_{c(i),k} := \frac{1}{2} k_c d_{i,k}^2 \tag{3.91}$$

where $k_c \in \Re^+$ is the contact spring, and $d_{i,k} := n_{i,k}^T (p_{w_i,k}^c - p_{i,k}^c) \in \Re$ is the penetration depth at i-th contact point (see Fig. 3-4). For this $\psi_{c(i),k}$ and $d_{i,k}$, we want an energy generation by the normal contact force $\lambda_{i,k}^n$ to be

$$\psi_{c(i),k+1} - \psi_{c(i),k} = \hat{V}_k^T \left({Ad_{i,k}^c}^T n_{i,k} \lambda_{i,k}^n \right) T_k \tag{3.92}$$

where $Ad_{i,k}^c$ is defined in (3.69). Then, with the Taylor expansion of $\psi_{c(i),k+1}$, we can get the passive penalty force $\lambda_{i,k}^n$, similar to the linear spring of (2.13), s.t.,

$$\lambda_{i,k}^n = -k_c \frac{d_{i,k} + d_{i,k+1}}{2} = -k_c \left(d_{i,k} + \frac{T_k}{2} n_{i,k}^T Ad_{i,k}^c \hat{V}_k \right) \tag{3.93}$$

with which the normal force part of $F_{i,k}^c$ in (3.69) is then given as

$$F_{i,k}^{c_n} = -k_c d_{i,k} {Ad_{i,k}^c}^T n_{i,k} - \frac{k_c T_k}{2} {Ad_{i,k}^c}^T n_{i,k} n_{i,k}^T Ad_{i,k}^c \hat{V}_k \tag{3.94}$$

where the first term is exactly in the same form of the explicit penalty force and the second term is damping-like term to passify the energy generation due to the discretization. This normal contact force is then decoded and integrated to the discrete-time dynamics (3.57), similar to the joint constraint force, which is solved without complex pivoting algorithm.

On the other hand, the friction force can be formulated in two ways: with elastic force and with constraint force. Friction force formulation with elastic force has same form with (3.94) except using $t_{i,k}^1, t_{i,k}^2$ rather than $n_{i,k}$. However, in this case, every contact points should be tracked to calculate the friction force. Other way to avoid this contact points tracking is to using constraint based friction force. For this, we

can rewrite the PMI simulation equation (3.74) as

$$\hat{V}_k^a = A_k^{-1}(B_k + \sum_{i\in\mathcal{C}} \sum_{j=1,2} S_{c(i)} {Ad_{i,k}^c}^T t_{i,k}^j \lambda_{i,k}^{t_j}) =: A_k^{-1}\left(B_k + \bar{D}_k \lambda_k^t\right) \tag{3.95}$$

where $S_{c(i)} \in \Re^{6(m+1)\times 6}$ is a selection matrix which applies each friction force to an appropriate contact body, and A_k and B_k are in (3.74) which contains (3.94). Then, the tangential velocity along $t_{i,k}^1, t_{i,k}^2$ can be written

$$\bar{D}_k^T \hat{V}_k^a = \bar{D}_k^T A_k^{-1} B_k + \bar{D}_k^T A_k^{-1} \bar{D}_k \lambda_k^t. \tag{3.96}$$

This (3.96) is similar to the second equation of (3.79). However, since here we do not use the pyramidal cone approximation, we do not have positive value constraint (i.e., no requirement on $\lambda_k^t \geq 0$). The objective of solving friction is equivalent to find λ_k^t which makes $\bar{D}_k^T \hat{V}_k^a$ of (3.96) be zero as much as possible while satisfying the original Coulomb friction constraint (i.e., $\mu|\lambda_k^n| \geq |\lambda_k^t|$). For this, we can employ the PGS-like iterative solver (see [13] for more details on that algorithm). Note that, we can also derive this penalty force for the Lagrange dynamics in generalized coordinate of Sec. 3.1, in a similar ways, by replacing/incorporating the adjoint operator with the Jacobian as in Sec. 3.3.1. Note that, though it is faster than (3.79)-(3.80), solving (3.96) for constraint-based friction can slow down the simulation when there are many contact points. For this reason, as mentioned early in this section 3.3.2, some clustering algorithm is required. In Chap. 4, we will introduce the data-driven learning-based approach for this clustering which tries to validate its accuracy from the experiment data.

3.4 Passive Virtual Coupling for PMI Simulation

In this section, we introduce the virtual coupling technique which is widely used in, especially the haptic rendering with the kinesthetic haptic device. The basic aim of haptic systems [3, 68] is to allow human users to physically interact with a

computer-simulated virtual environment, or virtual objects therein, through a robotic manipulator, which is called the haptic device, serving the roles of both measuring the human motion and generating force feedback for them. To connect the virtual world and the haptic device, many haptic interaction systems adopt the architecture as shown in Fig. 3-9, where the virtual proxy [69] (or, god-object [2, 7]) is an avatar representing the haptic device in the virtual world, and the virtual coupling defines a rule connecting the virtual proxy and the haptic device.

The virtual proxy is then simulated to follow the pose of the haptic device, while also respecting the physics of the virtual world, in particular, no-penetration constraint [2, 7, 70], energy-consistent penetration [64], and some desired dynamics of the virtual proxy itself [71, 4]. Force feedback to the kinesthetic haptic device is then typically computed as the force output of the spring-damper connection between the virtual proxy and the haptic device. In this thesis, we first introduce how to design the standard spring-damper type proxy-device virtual coupling in the PMI simulation and the condition to guarantee its passivity [4]. After then, we introduce the *transparent virtual coupling* which exploits the passivity of PMI simulation and the passive decomposition [26], and is able to overcome the limitation of the standard virtual coupling stemming from the compliant nature of it.

To design virtual coupling and analyse its passivity, here we assume that the haptic device (HD) and the virtual proxy (VP) is kinematically similar and has same motion DOFs. Thus, for the conventionally kinesthetic haptic device such as Touch$^{\text{TM}}$, the virtual proxy will be simulated via the PMI in generalized coordinate explained in Sec. 3.1. Then, we aim to couple the haptic device configuration $q_1(t)$ to the proxy configuration $q_{2,k}$ (corresponding to q_k in (3.14)). For this, we define the virtual coupling control $\tau_1(t) \in \Re^n$ (corresponding to f in (2.4)) for the haptic device s.t.: for $t \in T_k = [t_k, t_{k+1})$,

$$\tau_1(t) := -b_d \dot{q}_1(t) - B_{\text{vc}} \left(\frac{q_{1,k} - q_{1,k-1}}{T_{k-1}} - \hat{v}_{2,k-1} \right) - K_{\text{vc}} \left(q_{1,k} - q_{2,k} \right) \tag{3.97}$$

where $q_{1,k} \in \Re^n$ is the device configuration sampled at t_k, $b_d > 0$ is the mini-

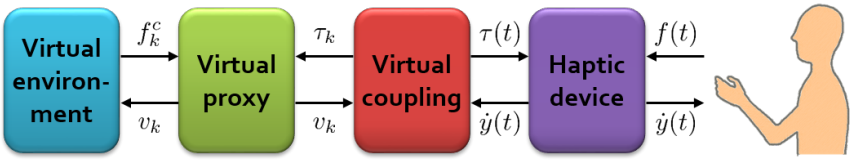


Figure 3-9: Block diagram of haptic interaction, where human users interact with virtual environments via the cascade of haptic device, virtual coupling, and virtual proxy.

mum device damping, $B_{vc}, K_{vc} \in \Re^{n\times n}$ are the virtual coupling gains, and $\hat{v}_{2,k-1} := (q_{2,k} - q_{2,k-1})/T_{k-1}$ are the VP velocity in (3.2). This $\tau_1(t)$ except $b_d\dot{q}_1(t)$ is computed at t_k and applied through t_{k+1}, while $b_d\dot{q}_1(t)$ is provided by the device.

Similarly, the virtual coupling control $\tau_{2,k}$ (corresponding to f_k in (3.14)) for the VP (3.14)-(3.17) is defined s.t., during $T_k = [t_k, t_{k+1})$,

$$\tau_{2,k} := -b_v\hat{v}_{2,k} - B_{\rm vc}\left(\hat{v}_{2,k} - \frac{q_{1,k} - q_{1,k-1}}{T_{k-1}}\right) - K_{\rm vc}(\hat{q}_{2,k} - q_{1,k}) \tag{3.98}$$

where $b_v > 0$ is the VP's extra damping to enforce passivity, $\hat{v}_{2,k} = (q_{2,k+1} - q_{2,k})/T_k$ and $\hat{q}_{2,k} = (q_{2,k+1} + q_{2,k})/2$. This $\tau_{2,k}$ is then plugged in the PMI simulation (3.14)-(3.17) and integrated to produce ξ_{k+1}, q_{k+1}.

Let us now analyse the energy generation by $(\tau_1(t), \tau_{2,k})$ during $T_k = [t_k, t_{k+1})$, which we want to be bounded to enforce passivity of the virtual coupling block in Fig. 3-9:
1) energy dissipation via the device and virtual damping (b_d, b_v):

$$\text{diss}(k) := -b_d\int_{T_k}||\dot{q}_1(t)||^2dt - \frac{b_v}{T_k}||\Delta q_{2,k}||^2 \le -\frac{b_d}{T_k}||\Delta q_{1,k}||^2 - \frac{b_v}{T_k}||\Delta q_{2,k}||^2 \tag{3.99}$$

where we use $\Delta\star_k := \star_{k+1} - \star_k$, $||\star||^2 = \star^T\star$ and the fact that

$$\sum_{i=1}^{n}\int_{T_k}\dot{q}_{1_i}^2(t)dt \ge \frac{\sum_{i=1}^{n}\Delta q_{1_i}(k)^2}{T_k} = \frac{||\Delta q_1(k)||^2}{T_k} \tag{3.100}$$

from the Cauchy-Schwartz inequality [72], where q_{1_i} is the i-th component of $q_1 \in \Re^n$.

2) energy generation by the virtual coupling damping $B_{\rm vc}$:

$$E_B(k) := -\left\langle \Delta q_{1,k}, \frac{\Delta q_{1,k-1}}{T_{k-1}} - \frac{\Delta q_{2,k-1}}{T_{k-1}} \right\rangle_{B_{\rm vc}} - \left\langle \Delta q_{2,k}, \frac{\Delta q_{2,k}}{T_k} - \frac{\Delta q_{1,k-1}}{T_{k-1}} \right\rangle_{B_{\rm vc}} \tag{3.101}$$

$$\leq \frac{||\Delta q_{1,k-1}||^2_{B_{\rm vc}}}{T_{k-1}} + \frac{||\Delta q_{1,k}||^2_{B_{\rm vc}}}{T_{k-1}} + \frac{||\Delta q_{2,k-1}||^2_{B_{\rm vc}}}{2T_{k-1}} - \frac{||\Delta q_{2,k}||^2_{B_{\rm vc}}}{2T_k} + \frac{1}{2}\left(\frac{1}{T_{k-1}} - \frac{1}{T_k}\right)||\Delta q_{2,k}||^2_{B_{\rm vc}} \tag{3.102}$$

where $\langle v, w\rangle_P := v^T P w$ and $||w||^2_P := \langle w, w\rangle_P$.

3) energy generation by the virtual coupling spring $K_{\rm vc}$:

$$E_K(k) := -\langle q_{1,k} - q_{2,k}, \Delta q_{1,k}\rangle_{K_{\rm vc}} - \langle \hat{q}_{2,k} - q_{1,k}, \Delta q_{2,k}\rangle_{K_{\rm vc}} \tag{3.103}$$

$$\leq \tfrac{1}{2}||\Delta q_{2,k}||^2_{K_{\rm vc}} + ||\Delta q_{1,k}||^2_{K_{\rm vc}} - \tfrac{1}{2}||\Delta e_{k+1}||^2_{K_{\rm vc}} + \tfrac{1}{2}||\Delta e_k||^2_{K_{\rm vc}} \tag{3.104}$$

where $\Delta e_k := q_{1,k} - q_{2,k}$.

Now, combining these for $k = 0, ..., N$, we have the following conditions for passive virtual coupling:

$$b_d I \succeq B_{\rm vc}\left(1 + \frac{T_k}{T_{k-1}}\right) + K_{\rm vc}T_k \tag{3.105}$$

$$b_v I \succeq \frac{B_{\rm vc}}{2}\left(\frac{T_k}{T_{k-1}} - 1\right) + \frac{K_{\rm vc}T_k}{2} \tag{3.106}$$

with which the hybrid-time passivity of the virtual coupling block in Fig. 3-9 is achieved:

$$\int_0^{\bar{t}} \tau_1^T(t)\dot{q}_1(t)dt + \sum_{k=0}^{N} \tau_{2,k}^T \hat{v}_{2,k} T_k \leq c^2 := \frac{1}{2}||\Delta e_o||^2_{K_{\rm vc}} \tag{3.107}$$

where $\bar{t} := t_{N+1}$ and $c \in \Re$. This passivity condition (3.105)-(3.106) implies that 1) the virtual damper and spring, $B_{\rm vc}, K_{\rm vc}$, can generate energy, with acting on the delayed terms in (3.97) and (3.98); and 2) this energy generation should be absorbed by the dampers, b_d and b_v. Here, the damper b_v dissipates energy even though it is a

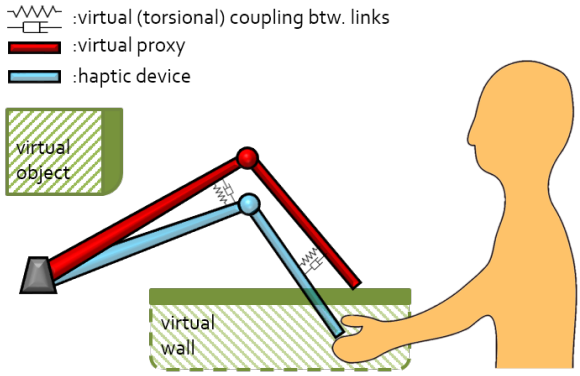


Figure 3-10: Degradation of transparency of standard virtual coupling during contact task due to device-proxy position deviation and unscaled inertias of the device and the proxy. Our proposed virtual coupling aims to render the haptic device and the virtual proxy as one single rigid tool with perfect device-proxy coordination and zero apparent device-proxy inertia via inertia scaling.

virtual damper, as it implemented via the PMI.

The passivity conditions (3.105)-(3.106) are similar to the well-known Colgate's passivity condition [24, 72], yet, more conservative (e.g., the first condition (3.105) is twice scaled-down in [24, 72], while the second condition (3.106) vanishes in [24, 72]). This is due to the implicit nature of the PMI. For example, instead of using $(\hat{v}_{2,k}, \hat{q}_{2,k})$, which are desired for the PMI's passivity (e.g., (3.16)) yet not available during $T_k = [t_k, t_{k+1})$, $(\hat{v}_{2,k-1}, q_{2,k})$ are used in $\tau_1(t)$. This results in the twice more conservative condition (3.105) than that in [24, 72]. On the other hand, the last term of the second condition (3.106) is induced since only the "delayed" $q_{1,k}$ is used in $\tau_{2,k}$, instead of the desired $\hat{q}_{1,k} = (q_{1,k+1} + q_{1,k})/2$, which then needs to be passified by the (discrete-time) damping b_v.

Now, suppose that a human user makes a contact with the virtual environment. Then, some device-proxy coordination error is inevitable, as the force feedback is generated by the virtual spring and damper $K_{\rm vc}, B_{\rm vc}$ as illustrated in Fig. 3-10. This coordination error then degrades the transparency of the haptic interaction (e.g., confused perception of the boundaries/positions of virtual objects). However, as suggested by the above discussion on the passivity condition (3.105)-(3.106), we cannot indefinitely increase $K_{\rm vc}$ to suppress this proxy-device deviation. This then means that, to cope with this problem, we would need to use some feedforward

action of the human and proxy forces. How to design such feedforward action is not so obvious for multi-DOF nonlinear haptic device and virtual proxy. In the following Sec. 3.4.1, we will utilize passive decomposition [26, 73] to do so, while also utilizing disturbance observer (e.g., [74, 75]) to estimate human interaction force.

3.4.1 Improving Transparency of Virtual Coupling

To overcome these limitations of the standard virtual coupling, we propose a novel virtual coupling framework for a multi-DOF haptic device and a dynamic virtual proxy. Our proposed virtual coupling, utilizing passive decomposition [26, 73] and a human force observer [74, 75], can maintain the device-proxy coordination even during the contact tasks, while also allowing for scaling down (or up) the apparent inertia of the coordinated device-proxy system, thereby, substantially improving the transparency of the standard virtual coupling. We also experimentally and theoretically show that, as long as the apparent inertia is not scaled down below a certain value (i.e., minimum-possible passive apparent inertia), our proposed virtual coupling can ensure passivity, particularly when used with the passive integration scheme of this PMI.

Let us start with briefly reviewing passive decomposition in the continuous-time domain, which will then be applied to the continuous-time haptic device and the discrete-time virtual proxy. Following [26], consider the dynamics of two n-DOF robots:

$$M_1(x_1)\ddot{x}_1 + C_1(x_1, \dot{x}_1)\dot{x}_1 = \tau_1 + f_1 \tag{3.108}$$

$$M_2(x_2)\ddot{x}_2 + C_2(x_2, \dot{x}_2)\dot{x}_2 = \tau_2 + f_2 \tag{3.109}$$

where the $x_i, \tau_i, f_i \in \Re^n$ are the configuration, the control torque (i.e., virtual coupling torque) and the external force (i.e., human force and virtual interaction force); and $M_i, C_i \in \Re^{n\times n}$ are the inertia and Coriolis matrices with positive-definite M_i and skew-symmetric $\dot{M}_i - 2C_i$. Here, we assume that the gravity has been locally canceled out or incorporated into the terms f_1, f_2.

Passive decomposition [26, 73] is then given by the following (non-singular) decomposition matrix:

$$S(x_1, x_2) := \begin{bmatrix} I - \phi(x_1, x_2) & \phi(x_1, x_2) \\ I & -I \end{bmatrix} \in \Re^{2n \times 2n} \tag{3.110}$$

where $\phi(x_1, x_2) := [M_1(x_1) + M_2(x_2)]^{-1} M_2(x_2)$. Using this, we can then transform the velocity and the control/external forces of (3.108)-(3.109) s.t.,

$$\begin{bmatrix} v_L \\ v_E \end{bmatrix} = S(x_1, x_2) \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}, \qquad \begin{bmatrix} \tau_L + f_L \\ \tau_E + f_E \end{bmatrix} = S^{-T}(x_1, x_2) \begin{bmatrix} \tau_1 + f_1 \\ \tau_2 + f_2 \end{bmatrix} \tag{3.111}$$

and the dynamics themselves (3.108)-(3.109) are also decomposed by

$$M_L(x)\dot{v}_L + C_L(x, \dot{x})v_L + C_{LE}v_E = \tau_L + f_L \tag{3.112}$$

$$M_E(x)\dot{v}_E + C_E(x, \dot{x})v_E + C_{EL}v_L = \tau_E + f_E \tag{3.113}$$

where $x := [x_1; x_2] \in \Re^{2n}$ and

$$\begin{bmatrix} M_L & 0 \\ 0 & M_E \end{bmatrix} := S^{-T} \begin{bmatrix} M_1 & 0 \\ 0 & M_2 \end{bmatrix} S^{-1} \tag{3.114}$$

with $M_L := M_1 + M_2$ and $M_E := \phi^T M_1 \phi + [\phi^T - I] M_2 [\phi - I]$ being the transformed positive-definite symmetric inertia matrices. The transformed Coriolis matrices are also similarly obtained s.t.,

$$\begin{bmatrix} C_L & C_{LE} \\ C_{EL} & C_E \end{bmatrix} := S^{-T} M \frac{d}{dt}(S^{-1}) + S^{-T} C S^{-1} \tag{3.115}$$

where $M := \text{diag}[M_1, M_2]$, $C := \text{diag}[C_1, C_2]$. From (3.115), we can also show that $C_{LE} = -C_{EL}^T$, that is, C_{LE} and C_{EL} define conservative (i.e., passive) power shuffling channels between the locked and shape systems.

Here, the dynamics (3.113) is called the *shape system*, which describes the coor-

dination error $x_E := x_1 - x_2$ with $v_E = \dot{x}_E = \dot{x}_1 - \dot{x}_2$ and with f_E being the "mismatched" portion of the coordination-perturbing human and proxy forces; whereas the dynamics (3.112) is called the *locked system*, which represents the overall coordinated dynamics of the two robots with $v_L = [M_1 + M_2]^{-1} \cdot [M_1\dot{x}_1 + M_2\dot{x}_2]$ and its inertia and external force respectively given by $M_L = M_1 + M_2$ and $f_L = f_1 + f_2$. For more details on passive decomposition, refer to [26, 73].

Aiming for the ideal transparency (i.e., perfect device-proxy coordination with zero apparent inertia), we design the following controls:

$$\tau_L := C_{LE}v_E + \tfrac{1-\eta}{\eta} f_L \tag{3.116}$$

$$\tau_E := C_{EL}v_L - B_{\text{vc}}v_E - K_{\text{vc}}x_E - f_E \tag{3.117}$$

where $\eta > 0$ is the inertia scaling factor to scale down (or up) the apparent inertia of the locked system M_L. Applying the controls (3.116)-(3.117) to (3.112)-(3.113), we can obtain the following closed-loop dynamics:

$$\eta \cdot [M_L(x)\dot{v}_L + C_L(x, \dot{x})v_L] = f_L \tag{3.118}$$

$$M_E(x)\dot{v}_E + C_E(x, \dot{x})v_E + B_{\text{vc}}v_E + K_{\text{vc}}x_E = 0 \tag{3.119}$$

where the second line indicates $x_E = x_1 - x_2 \to 0$ exponentially [73, 59] even during the contact tasks (i.e., perfect device-proxy coordination), whereas the first line shows that the coordinated device-proxy system will exhibit apparent inertia of ηM_L (i.e., inertia scaling), with $\eta \to 0$ required for the ideal transparency.

Using (3.111), we then decode the controls (3.116)-(3.117) into τ_1 and τ_2. Then by applying this to the virtual coupling (3.97)-(3.98) of the haptic device and the

virtual proxy, we can obtain the following transparent virtual coupling:

$$\begin{aligned}\tau_1(t) =& - b_d\dot{x}_1(t) - B_{\rm vc}(\frac{x_{1,k}-x_{1,k-1}}{T_{k-1}} - \hat{v}_{2,k-1}) - K_{\rm vc}(x_{1,k}-x_{2,k}) \\ &+ C_{EL,k}v_{L,k} + (I-\phi_k^T)C_{LE,k}v_{E,k} + (\tfrac{1-\eta}{\eta}I - \tfrac{1}{\eta}\phi_k^T)(f_{1,k}+f_{2,k}) + f_{2,k} \quad (3.120)\\ \tau_{2,k} =& - b_v\bar{v}_{2,k} - B_{\rm vc}(\hat{v}_{2,k} - \frac{x_{1,k}-x_{1,k-1}}{T_{k-1}}) - K_{\rm vc}(\hat{x}_{2,k}-x_{1,k}) \\ &- C_{EL,k}v_{L,k} + \phi_k^T C_{LE,k}v_{E,k} + \tfrac{1}{\eta}\phi_k^T(f_{1,k}+f_{2,k}) - f_{2,k} \quad (3.121)\end{aligned}$$

where $\phi_k, C_{EL,k}, C_{LE,k}$ are their respective continuous-time variables sampled at t_k, and $f_{1,k}, f_{2,k}$ are the human force and the proxy force sampled at $t = t_k$. In (3.120)-(3.121), similar to (3.108)-(3.109), we assume the gravity has been already canceled out locally or embedded in $f_{i,k}$.

Here, note that, in (3.120)-(3.121), the portion of standard virtual coupling (3.97)-(3.98) is passive under (3.105)-(3.106). The terms with C_{LE} and C_{EL} turn out not to pose any significant issue either, in terms of passivity due to their skew-symmetric relation $C_{LE} = -C_{EL}^T$. Totally unclear however is whether the remaining terms, which contain the force feedforward actions of the inertia scaling and the cancellation of f_E are passive or not. Their passivity will be experimentally and theoretically established in Sec. 3.4.2.

The proposed virtual coupling (3.120)-(3.121) can substantially reduce the device-proxy coordination error even during the contact tasks, while also scaling down (or up, resp.) the apparent inertia of the coordinated proxy-device system with $\eta < 1$ (or $\eta > 1$, resp.), thereby, significantly improving the transparency of the standard virtual coupling. To implement (3.120)-(3.121), we need the human force $f_{1,k}$ and the proxy force $f_{2,k}$, where the latter is readily available from the simulation, while the former is not, as most commercial haptic devices do not have force sensors on them. For this, we utilize the disturbance observer proposed in [74, 75], as summarized in the next part, along with a remedy to the issue of device saturation.

Human Force Observer

Consider the dynamics of a haptic device (3.108). The human force may then be considered as a disturbance $d \in \Re^n$ s.t.,

$$d = M(x)\ddot{x} + C(x, \dot{x})\dot{x} - \tau \tag{3.122}$$

where we omit the subscript for simplicity. Denote the estimate of this disturbance d by $\hat{d} \in \Re^n$, and, following [74], define an auxiliary variable $z \in \Re^n$ s.t.,

$$\hat{d} := z + p \tag{3.123}$$

where $p \in \Re^n$ is any function satisfying $\dot{p} = LM\ddot{x}$. As suggested in [75], here, we choose $p = L(x)M(x)\dot{x} = \alpha\dot{x}$ with $L(x) = \alpha M^{-1}(x) \in \Re^{n\times n}$ and $\alpha > 0$. The estimate $\hat{d}$ is then updated with the following update dynamics of z:

$$\dot{z} = -L(x)z + L(x) \cdot [C(x, \dot{x})\dot{x} - \tau - p] \tag{3.124}$$

Convergence of this human force observer can then be shown as follows. Define the force estimation error $e_d := d - \hat{d}$. We can then obtain the dynamics of e_d s.t.,

$$\dot{e}_d = \dot{d} - (\dot{z} + L(x)M(x)\ddot{x}) = \dot{d} - L(x)[M(x)\ddot{x} + C(x, \dot{x})\dot{x} - \tau - p - z] \tag{3.125}$$

$$= \dot{d} - L(x)[M(x)\ddot{x} + C(x, \dot{x})\dot{x} - \tau - \hat{d}] = \dot{d} - L(x)e_d \tag{3.126}$$

where we use the definitions of d in (3.122) and $\hat{d}$, the update equation (3.124), and the condition $\dot{p} = LM\ddot{x}$. This then implies that, by choosing α large enough w.r.t. $||\dot{d}||$, there will exist a finite time $t_{\text{ub}} \geq 0$ s.t., $||e_d(t)|| \approx 0$ for all $t \geq t_{\text{ub}}$, since 1) with $L(x) = \alpha M^{-1}(x)$, the error dynamics of e_d (3.125) is ultimately bounded; and 2) in typical haptic interaction, a human command (e.g., $||\dot{d}||$) does not change so quickly (e.g., see Fig. 3-15), thus, we can indeed set α (or observer bandwidth w_f) large enough to make the ultimate bound of e_d small enough. See Sec. 3.4.2.

Although we found this human force observer (3.124) adequately functioning dur-

ing our experiments, we also found that it exhibited "lurking" and "wind-up" phenomena, when the device control $\tau_1(t)$ becomes saturated. To better see this, consider a static contact. To make exposition simpler, consider also only one axis of the haptic device. Then, the real device control τ will be $\tau = \tau_m$ if $\tau_c \geq \tau_m$; or $\tau = \tau_c$ if $\tau_c \leq \tau_m$, where τ_c is the intended torque command and τ_m the maximum possible (saturated) control torque.

Now, suppose that τ is saturated by τ_m, yet, this is not properly reflected in the force estimation (3.124). Then, in steady-state, the estimated force will be $\hat{d} = -\tau_c$ (from (3.124)), yet, the real force is $d = -\tau_m$ (from (3.122)). Combining these two equations, we can then obtain

$$\hat{d} = d - (\tau_c - \tau_m) \tag{3.127}$$

showing that the observer $\hat{d}$ will over-estimate the real human force d by the magnitude of $\tau_c - \tau_m \geq 0$, increasing the feedforward terms in the virtual coupling (3.120)-(3.121). Here, note that, in static contact, the signs of d and τ are opposite.

Since the feedforward terms in (3.120)-(3.121) attempt perfect coordination and the device torque τ is already saturated, the haptic device will stay where it is, yet, the virtual proxy will start moving toward the haptic device. This then means that, during the static contact, the virtual proxy will "lurk" by itself, e.g., to the boundary of the virtual wall in Fig. 3-10. During this process, the estimated force $\hat{d}$ may also accumulate to a large value (via (3.124)). Then, when the human user tries to pull the haptic device out from the virtual wall, $\hat{d}$ may still be large, generating abnormally large pushing force from the virtual wall.

To prevent these "lurking" and "wind-up" phenomena, we choose to use τ_m instead of τ_c in the force observer (3.124) whenever $\tau_c \geq \tau_m$. This simple strategy turns out to be fairly effective in practice. This saturation problem was also considered in [75] by projecting $\hat{d}$ directly below τ_m (instead of τ).

Robustness against Inertial Parameter Uncertainty

Our proposed virtual coupling (3.120)-(3.121) requires inertial parameters of the haptic device, which are difficult to obtain exactly in practice. Our virtual coupling (3.120)-(3.121), however, turns out to be robust against this inertial uncertainty, as to be shown theoretically below and also experimentally in Sec. 3.4.2 (e.g., see Fig. 3-12). This robustness is not so obvious from (3.120)-(3.121) though, since many terms there, particularly those related to the force feedforward actions, will be affected by the uncertainty (e.g., $\phi, v_L, C_{LE}, C_{EL}$), hence becoming potentially harmful rather than helpful.

Now, to analyze the robustness of our virtual coupling against the inertial uncertainty, let us start with the decomposed dynamics (3.112)-(3.113). Then, we can obtain the following perturbed locked-shape dynamics:

$$\begin{pmatrix} M_L\dot{v}_L + C_L v_L + C_{LE} v_E \\ M_E\dot{v}_E + C_E v_E + C_{EL} v_L \end{pmatrix} = S^{-T}\hat{S}^T \begin{pmatrix} \tau_L \\ \tau_E \end{pmatrix} + \begin{pmatrix} f_L \\ f_E \end{pmatrix} \tag{3.128}$$

where $\hat{S}$ and $S^{-T}\hat{S}^T$ are given by

$$\hat{S} := \begin{bmatrix} I - \hat{\phi} & \hat{\phi} \\ I & -I \end{bmatrix}, \quad S^{-T}\hat{S}^T = \begin{bmatrix} I & 0 \\ \phi^T - \hat{\phi}^T & I \end{bmatrix} \tag{3.129}$$

with $\hat{\star}$ representing the term $\star$ under the uncertainty.

From (3.128) with (3.116), we can then see that the inertia scaling control will almost retain its effectiveness even under the inertial uncertainty, since the perturbed locked system dynamics in this case is given by

$$\eta \cdot [M_L\dot{v}_L + C_L v_L + (C_{LE} - \hat{C}_{EL})v_E] = f_L \tag{3.130}$$

where $f_L = f_1 + f_2$ does not suffer from the inertial uncertainty and $||v_E||$ is typically not so large during the experiments (see Fig. 3-12). From (3.128) with (3.117), we

can also obtain the perturbed shape system dynamics s.t.,

$$M_E\dot{v}_E + C_E v_E + b_{vc} v_E + k_{vc} x_E + (C_{EL} - \hat{C}_{EL}) v_L = \tfrac{1}{\eta}\tilde{\phi}^T f_L \tag{3.131}$$

where $\tilde{\phi} := \phi - \hat{\phi}$ and we use $\frac{1-\eta}{\eta}\tilde{\phi} f_L - \hat{f}_E + f_E = \frac{1}{\eta}\tilde{\phi}^T f_L$ with $\hat{f}_E = \hat{\phi}^T(f_1 + f_2) - f_2$. However, during contact tasks, we have $\dot{x}_i \to 0$, which, from the above locked system dynamics (3.130), means that $f_L \to 0$. This then in turn implies from (3.131) that, with $\hat{f}_E \to f_E$ and $(\dot{v}_E, v_E) \to 0$, perfect device-proxy coordination will be guaranteed (i.e., $x_E \to 0$) even under the uncertainty.

Note from (3.97)-(3.98) and (3.120)-(3.121) that our proposed virtual coupling can be implemented on top of the (already-coded) standard virtual coupling. Also, as can be seen from (3.120)-(3.121), the structure of our proposed virtual coupling is rather straightforward and amenable to real-time implementation, only with the first-order human force observer (3.124) and the static feedforward/feedback control actions. The implementation of our virtual coupling (3.120)-(3.121) will yet become involved if the haptic device has large-DOF, since, in this case, some terms in our virtual coupling (3.120)-(3.121) (e.g., $\phi_k = [M_{1,k} + M_{2,k}]^{-1} M_{2,k}$) and also in the human force observer (3.124) (e.g., $L(x) = M^{-1}(x)$) can become high-dimensional/complicated. Recall though that: 1) our proposed virtual coupling possesses the robustness against this inertial uncertainty as shown above; and also 2) typical (or commercial) haptic devices have rather limited-DOF with their inertial parameters also often known. How to extend our proposed virtual coupling for haptic devices with large-DOF, possibly utilizing some iterative algorithms to handle the aforementioned high-dimensional and complex terms, is a topic for future research.

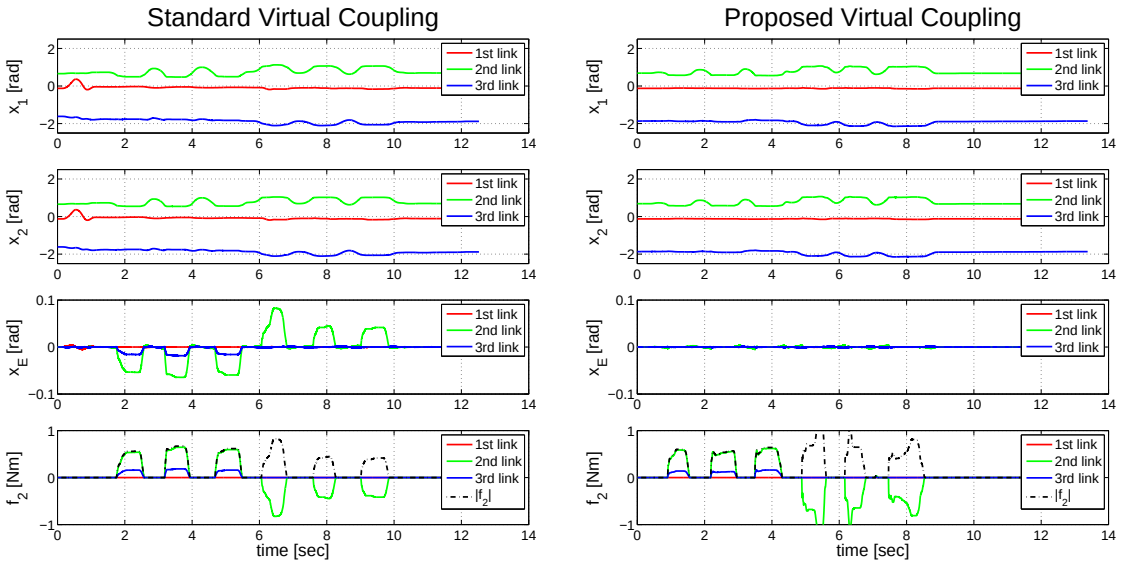
3.4.2 Performance Evaluation of Transparent Virtual Coupling

We implement our proposed virtual coupling (3.120)-(3.121) on the actuated 3-DOF of a Touch$^{\text{TM}}$, whose inertial parameters are obtained via least-square identification. We also render a three-DOF virtual proxy, kinematically similar to the haptic device.

We allow this virtual proxy to make multi-point/intra-link contacts, that is, on the middle of its proximal link and also at the end-effector of the distal link. See Fig. 3-10. Both of these intra-link and end-effector contacts are rendered as pure stiffness contacts with no damping dissipation, with their respective stiffness set by $K_{\text{proximal}} = 20$[Nm/rad] and $K_{\text{end-tip}} = 600$[N/m]. The PMI integration of Sec. 3.1 is utilized to simulate the multi-DOF nonlinear dynamics of the virtual proxy and the linear elastic dynamics of the virtual objects. Due to its enforcing discrete-time passivity, this PMI integration enables us to choose a very small inertia of the virtual proxy (and also very large stiffness if the contact is sustained). We choose the inertia of the virtual proxy to be of the order of around 10^{-4} w.r.t. the inertia of the haptic device. The device servo-rate and the virtual world simulation rate are both $T_k = 1$[ms].

For the standard virtual coupling (3.97)-(3.98), we set its gains to be $K_{\text{vc}} = 10$[Nm/rad] and $B_{\text{vc}} = 0.001$[Nm s/rad]. To choose these gains, we increase K_{vc} until the device starts to vibrate, which corresponds to the first passivity condition (3.105). We also set $b_v = 0.012$[Nm s/rad] to be around twice larger than that as required by the second passivity condition (3.106). We also empirically optimize the human force observer gain to be $\alpha = 0.035$ with the performance-passivity/robustness trade-off (i.e., set α as high as possible for the performance (cf. (3.125)), yet, not so high to passively attain the desired scaled-down inertia (cf. (3.137)) and also robustness against noise, etc.) and found its bandwidth to be $w_f \approx 200$[rad/s].

In the first experiment, we compare the performance of the standard virtual coupling (3.97)-(3.98) with that of our proposed virtual coupling (3.120)-(3.121). A human user makes three successive contacts on the proximal link, and then another three contacts at the end-effector of the 3-DOF virtual proxy by manipulating the 3-DOF haptic device. The user tries his/her best to make the haptic interaction as similar as possible between the two virtual coupling schemes. The results are presented in Fig. 3-11, where we can see that our proposed virtual coupling (3.120)-(3.121) can maintain very good proxy-device coordination even during the contacts, while the standard virtual coupling (3.97)-(3.98) exhibits substantial device-proxy coordination error during the contacts. This significantly-enhanced coordination per-



(a) Standard virtual coupling (3.97)-(3.98)

(b) Proposed virtual coupling (3.120)-(3.121)

Figure 3-11: Comparison of the standard virtual coupling (3.97)-(3.98) with the proposed virtual coupling (3.120)-(3.121) during 3-DOF haptic interaction with intermittent contacts, the first three on the middle of the proximal link and the last three on the end-effector. Notice our proposed virtual coupling maintains the device-proxy coordination even during the contacts, whereas the standard virtual coupling does not.

formance of our proposed virtual coupling can even be better seen from Fig. 3-12, where, for the standard virtual coupling, the coordination error is proportional to the contact force, while, for our proposed virtual coupling, the coordination error is bounded and remains small regardless of the magnitude of the contact force.

The second experiment is performed to show the efficacy of our inertia scaling control with $\eta = \{0.6, 1, 5\}$. For this, the human user manipulates the 3-DOF haptic system to incur a temporally/spatially-similar free motion of the virtual proxy with the different inertia scaling factors. The results are shown in Fig. 3-13, where we can see that the human force, required to produce this spatially/temporally-similar free motion and estimated by our human force observer, gets smaller with $\eta = 0.6$ (i.e., apparent inertia scaled-down) and larger with $\eta = 5$ (i.e., apparent inertia scaled-up) as compared to the case of $\eta = 1$ (i.e., no apparent inertia scaling). The results in Fig. 3-13 together with that in Fig. 3-11 then clearly indicate that, with $\eta = 0.6$, our

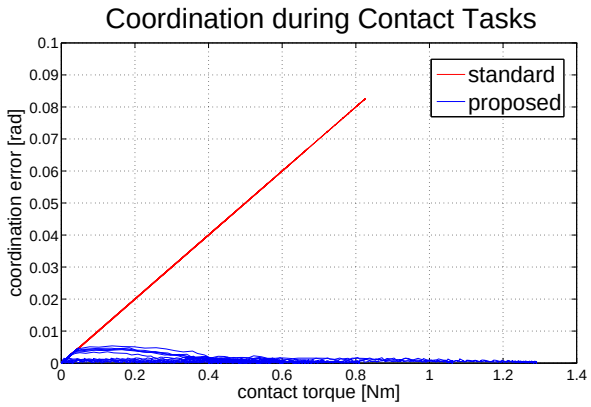


Figure 3-12: Device-proxy coordination performance of the standard virtual coupling and the proposed virtual coupling. Our proposed virtual coupling can maintain the coordination regardless of contact force, whereas coordination error of the standard virtual coupling is proportional to contact force.

proposed virtual coupling (3.120)-(3.121) substantially improves the transparency of the standard virtual coupling (3.97)-(3.98), with much less intermediate impedance between the haptic device and the virtual proxy in Fig. 3-9.

Finally, the third experiment is conducted to experimentally show that our proposed virtual coupling (3.120)-(3.121), even with the force feedforward action, still enforces passivity when η is not so small. For this, a human user performs a succession of free motion, contact with the wall and circle-drawing on the contact surface. The user carries out 10 trials of this composite scenario with $\eta \in \{0.6, 1, 3, 5\}$ and also with the standard virtual coupling. The user also controls the operation time to be similar among these trials so that we can compare their time-averaged passivity property as shown in Fig. 3-14. For this, since we do not have a human force sensor (although we do have a human force observer), we compute **controller passivity** [26], i.e.,

$$E_{c,NT} := \sum_{k=0}^{N} [\tau_{1,k} \cdot (x_{1,k+1} - x_{1,k}) + \tau_{2,k} \cdot (x_{2,k+1} - x_{2,k})] \leq 0, \quad \forall N \geq 0 \qquad (3.132)$$

where $T = 1$[ms] is the sampling rate, and $\tau_{1,k}$ is the zero-order-hold value of the haptic device control $\tau_1(t)$ at the time-instance kT. As shown in [26], this controller passivity implies the (closed-loop) passivity of our proposed virtual coupling. We

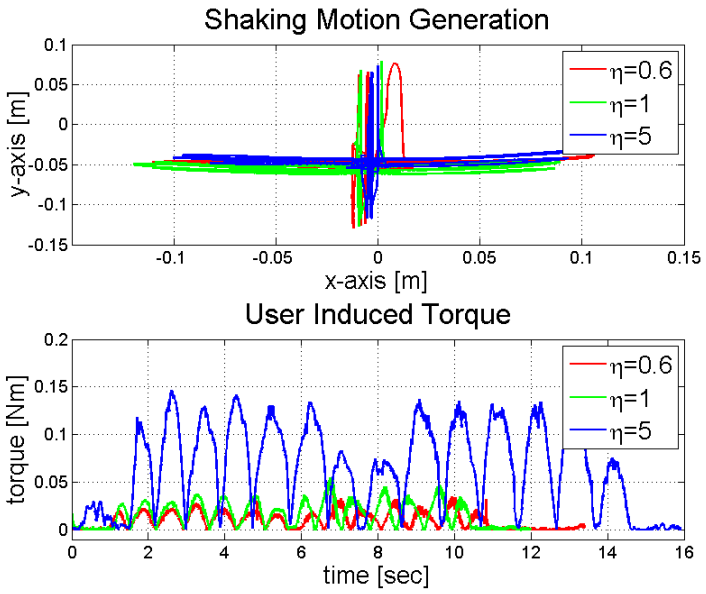


Figure 3-13: Comparison of human forcing to generate spatially/temporally-similar free motion with three different inertia scaling factors. Notice that the human force gets smaller with $\eta = 0.6$ (i.e., inertia scaled-down) and larger with $\eta = 5$ (i.e., inertia scaled-up) as compared to $\eta = 1$ (i.e., no inertia scaling).

can then see from Fig. 3-14 that: 1) standard virtual coupling under the passivity condition (3.105)-(3.106) is indeed passive; 2) our proposed virtual coupling is less passive than the standard virtual coupling if $\eta < 1$ and more passive if $\eta > 1$; and 3) our proposed virtual coupling is passive even with the force feedforward action, if η is not so small (i.e, becomes not passive with $\eta \leq 0.5$: not shown here).

In the next part, we elucidate why the inertia scaling η affects the passivity of our proposed virtual coupling in this way, i.e., why a smaller inertia scaling $\eta < 1$ reduces the "margin" of passivity (i.e., margin from the zero-line in Fig. 3-14) and why the scaled-up inertia (i.e., $\eta > 1$) still enforces passivity even if the human/virtual force feedforward actions are present in (3.120)-(3.121) with their effect getting more intensified as η increases from $\eta = 1$ to $\eta \to \infty$.

Passivity Analysis with Inertia Scaling η

The unabridged passivity analysis of our proposed virtual coupling with the multi-DOF haptic device and the virtual proxy is excessively difficult, or more likely, im-

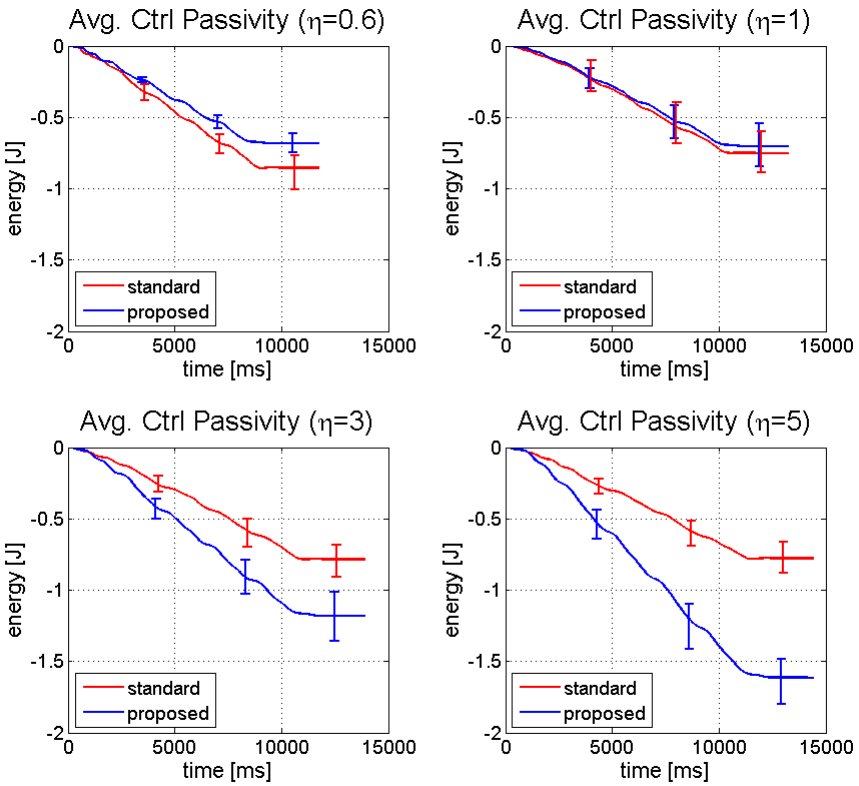


Figure 3-14: Controller passivity E_c of the proposed virtual coupling and the standard virtual coupling (averaged for 10 trials with minimum/maximum indicated too). Notice that inertia scaling down (i.e., $\eta < 1$) reduces "margin" of passivity, whereas inertia scaling up (i.e., $\eta > 1$) increases this margin, implying that our proposed virtual coupling is passive if η is not so small.

possible, since it is a nonlinear, hybrid, and multi-input-multi-output system with arbitrary forcing (f_1, f_2) and such complex issues as Coulomb friction, sensing uncertainty/quantization, etc. To make our analysis feasible while still retaining essential relation between the inertia scaling η and passivity, we make the following simplifying assumptions motivated by observation of the system behavior during our experimental studies.

We first assume that the dominant effect of the inertia scaling on passivity can be captured by the continuous-time dynamics, i.e., similar to (3.128), we utilize the

following dynamics for our analysis here:

$$\begin{pmatrix} M_L\dot{v}_L + C_L v_L + C_{LE} v_E \\ M_E\dot{v}_E + C_E v_E + C_{EL} v_L \end{pmatrix} = \begin{pmatrix} \tau_L + f_L \\ (\phi^T - \hat{\phi}^T)\tau_L + \tau_E + f_E \end{pmatrix} \tag{3.133}$$

with $\tau_L = \hat{C}_{LE} v_E + \frac{1-\eta}{\eta}\hat{f}_L = \hat{C}_{LE} v_E + \frac{1-\eta}{\eta}(\hat{f}_1 + \hat{f}_2)$ and $\tau_E = \hat{C}_{EL}\hat{v}_L - b_{vc} v_E - k_{vc} x_E - \hat{f}_E$, where $\hat{\star}$ is $\star$ with the inertial uncertainty or the force observer inaccuracy. This simplification, we believe, is adequate, since, here, we aim to analyze passivity as related to η, not related to the sampling or other discrete-time phenomena.

We further make the following assumptions, which are also motivated by our observation of the experimental results in Fig. 3-11 - Fig. 3-14:

- The shape system dynamics of v_E is not dominant on the relation between passivity and η, as we observed that x_E remains bounded (see Fig. 3-12) and behaves similarly during all the experiments in Fig. 3-14, implying that the most prevailing factor on passivity in Fig. 3-14 is η (i.e., locked system dynamics).

- The effects of C_L, C_E, C_{LE}, C_{EL} are all negligible, as the operations in Fig. 3-14 are not so fast.

- $\phi \approx \hat{\phi}$, since, in $\phi = [M_1 + M_2]^{-1} M_2$, the device inertia M_1 is the only uncertain term, and the virtual proxy inertia M_2 is set much smaller than M_1 as stated in the beginning of Sec. 3.4.2.

- $\hat{F}_L(s) \approx \frac{w_f}{s+w_f} F_L(s)$, where $s \in \mathbb{C}$ is the Laplace variable and $w_f \geq 0$ is the cut-off frequency of the human force observer (3.125). We make this assumption here even though $\hat{f}_L = \hat{f}_1 + \hat{f}_2$, as we observed from our experiments that non-passive oscillation occurs typically during the free-motion (i.e., with $f_2 \approx 0$), not during the contacts, for which the system behavior becomes much less oscillatory (i.e., likely more passive).

With these simplifying assumptions, the closed-loop dynamics (3.133) is reduced

to:

$$m_L\ddot{x}_L + b_L\dot{x}_L = \tfrac{1-\eta}{\eta}\hat{f}_L + f_L \tag{3.134}$$

where, for simplicity, we also assume that the locked dynamics in (3.133) is a scalar system with $m_L := m_1 + m_2 > 0$, $x_L := (m_1x_1 + m_2x_2)/m_L \in \Re$ with $v_L = \dot{x}_L$, and $b_L > 0$ is the damping stemming from b_d and b_v of (3.120)-(3.121). Then, applying the Laplace transform, we have

$$V_L(s) = \frac{1}{m_Ls + b_L} \cdot \left(1 + \frac{1-\eta}{\eta}\frac{w_f}{s+w_f}\right) F_L(s) =: H_L(s)F_L(s) \tag{3.135}$$

suggesting that passivity can be argued if and only if the transfer function $H_L(s)$ is positive-real (PR), i.e., its Nyquist plot lies in the RHP (right-half plane) in the complex domain [76].

Now, to see when the transfer function $H_L(s)$ is PR, we compute $H_L(jw)$ and check when its real-component is non-negative, i.e., $\mathrm{Re}[H_L(jw)] \geq 0\ \forall w \in [0, +\infty)$, where $j := \sqrt{-1}$. It is then easy to show that this is equivalent to:

$$R_L(w) := w_f^2 + \frac{\eta(w_c + w_f) - w_f}{w_c}w^2 \geq 0, \quad \forall w \in [0, +\infty) \tag{3.136}$$

where $w_c := \frac{b_L}{m_L}$ and w_f are the bandwidths of the open-loop locked system dynamics (3.134) and the human force observer (3.125). We can then show that $H_L(s)$ is PR iff

$$\eta \geq \frac{w_f}{w_c + w_f} \tag{3.137}$$

from which we can extract the following observations in concert with the experimental results in Fig. 3-11 - Fig. 3-14:

- Even though the force feedforward actions are utilized, the apparent inertia scaling-up (i.e., $\eta \geq 1$) is always passive, whereas passively scaling-down the apparent inertia (i.e., $\eta < 1$) is lower-bounded by a certain minimum-possible

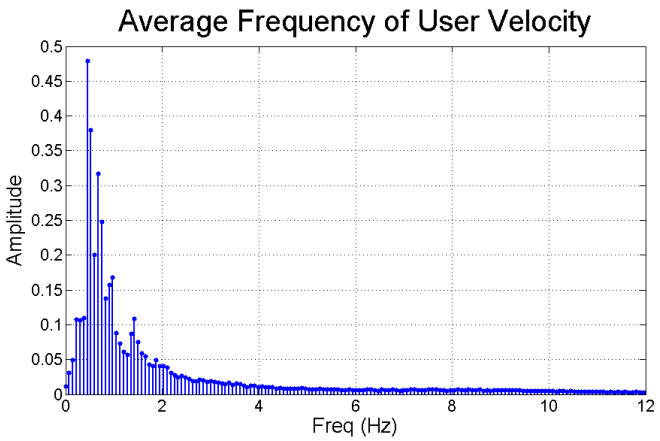


Figure 3-15: Frequency content of human motion during the experiments in Fig. 3-14 averaged over time. Note that the frequency content is mostly limited below 5Hz.

passive inertia scaling $\eta_{\min} := w_f/(w_c + w_f)$ (see Fig. 3-14).

- Larger damping b_L or smaller inertia m_L of the open-loop locked system (3.134) will allow for further reduction of the minimum-possible passive inertia scaling $\eta_{\min}$, as, in this case, we have more "passifying" damping effect and/or need less inertia scaling-down control action with m_L already small.

- A slower human force observer is more favorable to passively "include" smaller inertia scaling η, but not to necessarily "achieve" this intended η (i.e., passivity is enforced, yet, the inertia scaling η itself may be incompetent and not fully effective).

For our experimental setup, we have $w_f \approx 200$[rad/s] as stated in the beginning of Sec. 3.4.2. We also estimate $w_c = b_L/m_L \approx b_d/m_1 \approx 50$[rad/s], since: 1) for Touch$^{\text{TM}}$, we can estimate $b_d \approx 1.912$[Ns/m] by using the Colgate's passivity condition [24] and also have its apparent inertia to be $m_1 \approx 45$[g] according to the manufacturer's website[1]; and 2) the inertia and damping of the virtual proxy are set to be much smaller than those of Phantom Omni, thus, their contribution to m_L, b_L is rather negligible.

If we use the condition (3.137), we can then estimate $\eta_{\min} \approx 0.8$, which is heavier

[1] http://www.geomagic.com/en/products/phantom-omni/specifications

than $\eta = 0.6$ of our experiments in Fig. 3-14. Among many possible sources of this conservatism (e.g., unmodeled phenomena), a particularly interesting one is that our passivity analysis here is based on the assumption that non-passive behavior can arise for *any* frequency $w \in [0, +\infty)$, yet, in practice, human motions are only of limited bandwidth with perception-action asymmetry [77]. For instance, see Fig. 3-15 for the (averaged) frequency content of human motions during the experiments of Fig. 3-14, where the frequency content is limited mostly below 5Hz. This then means that we need to check the PR condition of $H_L(s)$ *only* for this limited bandwidth, not for all $w \in [0, +\infty)$. This may then allow us to substantially reduce the conservatism of the condition (3.137). How to estimate the minimum-possible passive apparent inertia, while incorporating human perception-action bandwidths (and other un-modeled phenomena) is a topic for future research. See [28, 78] for some related results along this direction.

Recall that our passivity analysis performed here relies on several assumptions made above. This then means that our analysis is not without restriction and should be applied with caution for cases where those assumptions are not granted, for instance: 1) the sampling rate is slow so that its effect on passivity is non-negligible; 2) the device-proxy coordination is compliant so that the shape system x_E dynamics becomes important; 3) the haptic interaction is so fast that the Coriolis effect becomes prominent; 4) the human force observer is of higher-order, or with dc-gain error and delay so that the first-order model above becomes invalid; and 5) virtual proxy inertia M_2 cannot be made arbitrarily small or major modes of instability occur during contacts (e.g., not using the PMI proposed in this thesis).

It is also worthwhile to mention that our passivity analysis here: 1) can be easily extended to the apparent inertia scaling problem of general human-interactive robots; and 2) explains why it is possible to completely eliminate any virtual energy storage elements in our proposed virtual coupling (under the condition (3.137)), which are in general deemed necessary in telerobotics/haptics to passively produce any force feedforward actions (e.g., [26, 79]).

3.5 Illustrative Examples of PMI-Based Haptic Rendering

So far, we have derived the PMI-based simulation framework including multi-point contact, which, due to its enforcing passivity effective in practice, can stably simulate wide-range of parameters (e.g., very light/stiff objects with no damping) regardless of (possibly varying/slow) integration steps T_k, that can occur during, e.g., intermittent multi-point collisions among virtual objects. To validate and demonstrate these (and other) advantages of our proposed PMI-based simulation framework, here, we perform three illustrative examples, namely, haptic operation of peg-in-hole task (Sec. 3.5.1), haptic interaction with flexible beam (Sec. 3.5.2), and multi-point power grasping of multi-fingered under-actuated tendon-drive (UATD) hand (Sec. 3.5.3).

3.5.1 Peg-in-Hole Task

For the peg-in-hole task, we model the peg and the box with holes as rigid bodies, whereas the two multi-fingered virtual hand as articulated rigid bodies with joint constraints as shown in Fig. 3-3, each composed of ten articulated rigid bodies. To simulate these, we apply the PMI-based formulation in maximal coordinates of Sec. 3.2. We also utilize the wearable cutaneous haptic interface proposed in [80], which can provide 3-DOF contact force feedback at the finger-tip. This cutaneous haptic device is different from typical haptic devices (e.g., Omega 3, Geomagic Phantom), as it cannot generate kinesthetic force feedback. See Chap. 5 and [80] for more details on this wearable cutaneous haptic device. Consequently, we utilize the unilateral virtual coupling to allow the user to command the position/orietation of the virtual hands/fingers, whereas the finger-tip contact force is directly fed back to the cutaneous haptic device.

More precisely, the SE(3) springs, consisting of E(3) linear spring of Sec. 2.2 and SO(3) spring of Sec. 3.2.1, are used as uni-directional position/orientation-commanding virtual coupling (e.g., [72, 81]). On the other hand, the contact force of

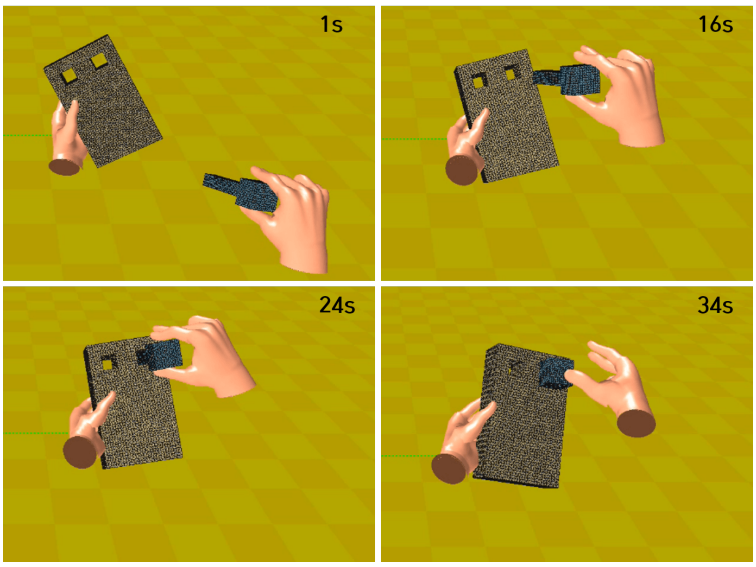


Figure 3-16: Snapshots of peg-in-hole task with two multi-fingered hands, each modeled as articulated rigid bodies as shown in Fig. 3-3 and making contact with the objects via friction contact simulated by PMI-LCP of Sec. 3.3.1.

each finger-tip of the virtual hands during the manipulation (computed via PMI-LCP of Sec. 3.3.1) is directly fed back to the cutaneous finger-tip haptic device. For this, we use

$$f_{f,k} = R_{f,k}^T H_{\text{filter}}(f_{i,k-1}^c, f_{i,k-2}^c, ...f_{i,k-p_f}^c) \tag{3.138}$$

where $R_{f,k}$ is the rotation matrix of the finger-tip body frame w.r.t. the inertial frame, and H_{filter} is a certain low-pass-filter to process the p_f-number of the computed contact force data $f_{i,k-j}^c$, $j = 1, ..., p_f$. The filter H_{filter} is used here to provide smooth cutaneous haptic feedback $f_{f,k}$ (with 1kHz) from the computed contact force $f_{i,k}^c$, whose update-rate is varying between 30-300Hz due to the simulation of the large number of rigid bodies (i.e., two hands, each composed of 10 rigid bodies, one box with holes, and one peg) and multi-point contacts among them. Note that we cannot use $f_{i,k}^c$ for H_{filter}, since it can be determined only after $T_k = [t_k, t_{k+1})$. The computational-lag from H_{filter} and the uni-laterality of the virtual coupling do not affect stability, since the device-human coupling here is only cutaneous, not kinesthetic [82].

Snapshots of this peg-in-hole task are shown in Fig. 3-16 with its finger-tip forces

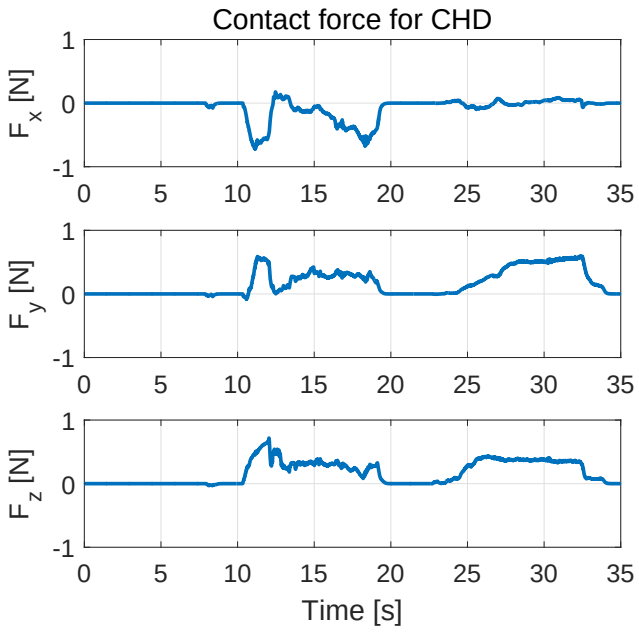


Figure 3-17: Finger-tip contact force during the peg-in-hole task of Fig. 3-16 expressed in the finger-tip coordinate frame (with z-axis being the contact normal direction).

also presented in Fig. 3-17. We set the mass/inertia of rigid bodies of the hand as $m = 0.001$[kg] and $J = mI_{3\times3}$[kgm^2], while the stiffness of the spring stiffness as $k_\lambda = 100$[N/m], with the stiffness/mass ratio to be fairly high of 10^5. We found that, with these parameters, the cutaneous haptic feedback rendering is transparent enough, since: 1) the virtual hand can (unilaterally) follow the user command fast enough with this high stiffness/mass ratio of 10^5; and 2) the cutaneous force feedback $f_{f,k}$, as computed above, is directly displayed to the finger-tip with no intermediate compliance. Note that this differs from the case of kinesthetic devices with bilateral virtual coupling, where, for the transparency, the virtual coupling should typically be much higher, as it defines bilateral compliant coupling between the virtual proxy and the real haptic device. Of course, we can also adopt kinesthetic haptic devices here along with, e.g., virtual coupling developed for the PMI [50, 4], or passive set-position modulation (PSPM: [25]). In this case, even if we use the PMI to enforce the passivity of the virtual worlds, the (hybrid/bilateral) virtual coupling gain will be still limited by the device physical damping - see [50].

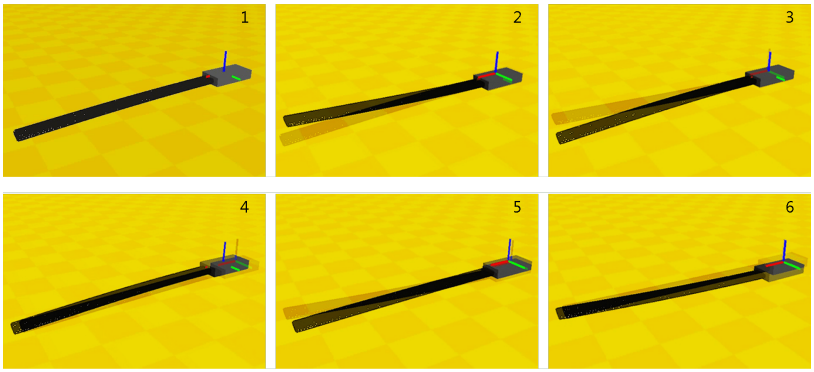


Figure 3-18: Snapshots of PMI-based simulation of undamped flexible beam harmonic oscillation.

3.5.2 Haptic Interaction with Flexible Beam

We model the flexible beam to be an aluminum beam with the dimension of (0.907, 0.05, 0.012)[m] and the mass of 1.4683[kg]. We also model its dynamics by using nodal coordinates with three natural frequencies of (11.91, 74.65, 209.03)[Hz]. Here, for simplicity, we assume the beam motion to be planar. Then, the beam dynamics can be modeled in mixed maximal-generalized coordinates and using Euler-Bernoulli model ([83]) s.t.,

$$\begin{bmatrix} M_{xx} & M_{x\delta} \\ M_{\delta x} & M_{\delta\delta} \end{bmatrix} \begin{pmatrix} \ddot{x} \\ \ddot{\delta} \end{pmatrix} + \begin{bmatrix} C_{xx} & C_{x\delta} \\ C_{\delta x} & 0 \end{bmatrix} \begin{pmatrix} \dot{x} \\ \dot{\delta} \end{pmatrix} + d\psi^T(\delta) = f \tag{3.139}$$

where $x \in \Re^3$ is the maximal coordinate of the beam end in SE(2), $\delta \in \Re^3$ is the generalized coordinate representing each vibration mode, $d\psi^T(\delta) := K_q q := [0; K_{\delta\delta}\delta]$, where $M_{\delta\delta}, K_{\delta\delta} \in \Re^{3\times3}$ are the constant inertia and stiffness matrices of vibration. On the other hand, other matrices $M_\star, C_\star \in \Re^{3\times3}$ are configuration-dependent nonlinear inertia and Coriolis matrices, with the off-diagonal terms representing the dynamic coupling.

Snapshots and profiles of kinetic, potential and total energies of undamped vibration of this flexible beam simulated by our PMI-based framework with $x = 0$ are shown in Fig. 3-18 and Fig. 3-19, respectively, where the integration interval is 1kHz. There, we can see that the total energy is conserved throughout this fairly

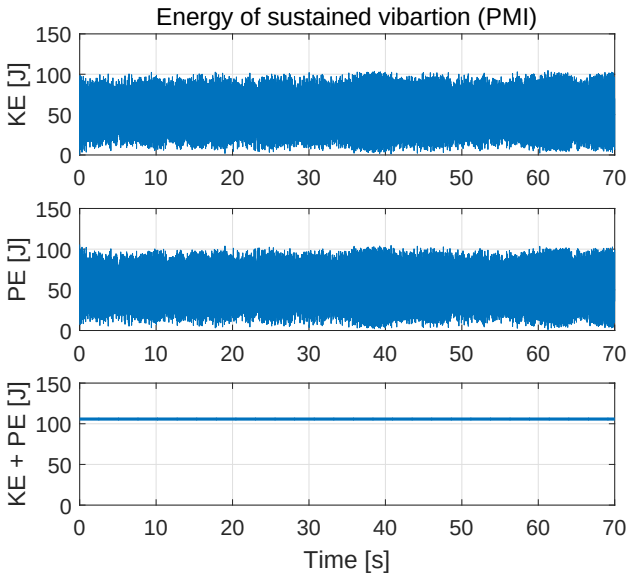


Figure 3-19: Profiles of kinetic, potential and total energies of the PMI-based simulation of undamped flexible beam vibration in Fig. 3-18: note the exact conservation of total energy during the fast oscillations.

fast undamped vibration, clearly showing the ability of our PMI-based framework to simulate lossless/marginally-stable/energy-conserving behaviors. This undamped harmonic oscillation cannot be stably simulated by the SEI. It can however be done by the IEI, for which we must have adopted some linearization as introduced in [13, 15] to render the IEI interactively-fast. The results with this IEI simulation are shown in Fig. 3-20, where it is clear that the IEI exhibits strictly-stable oscillation with the total energy quickly decaying to zero. This lack of ability to simulate lossless behaviors of the IEI is in fact due to its dissipativity (see Sec. 2.2.1), meaning that its advantage over our PMI framework is unclear, as the implementation complexity is at least comparable between PMI and IEI.

For the haptic interaction, we connect the flexible beam to a square rigid box via a spring, which is in turn connected to Omega 3® haptic device via virtual coupling

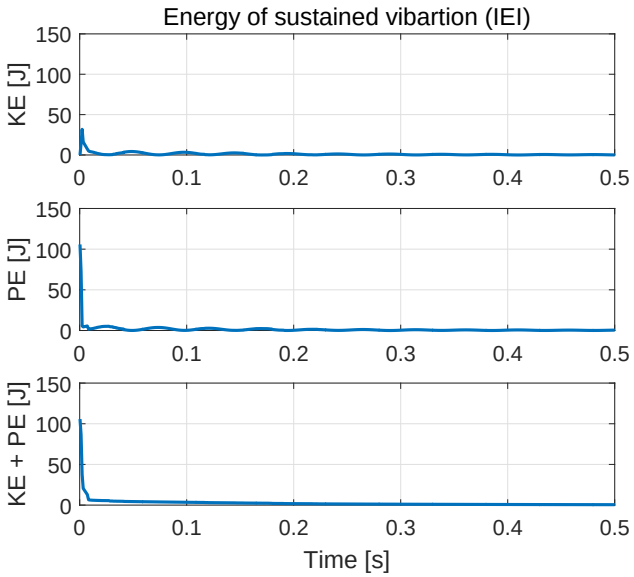


Figure 3-20: Profiles of kinetic, potential and total energies of the IEI-based simulation of undamped flexible beam vibration in Fig. 3-18 with linearized inertia matrix: note the total energy quickly decays due to dissipativity of the IEI, although the same initial conditions as that of Fig. 3-19 are used.

([4]). See Fig. 3-21. The dynamics of the total system can then be written by

$$\begin{bmatrix} M_r & 0 \\ 0 & M(q) \end{bmatrix} \begin{pmatrix} \ddot{x}_r \\ \ddot{q} \end{pmatrix} + \begin{bmatrix} B_r & 0 \\ 0 & C(q) \end{bmatrix} \begin{pmatrix} \dot{x}_r \\ \dot{q} \end{pmatrix} + \begin{bmatrix} K_r & -K_{rq} \\ -K_{rq}^T & K_q \end{bmatrix} \begin{pmatrix} x_r \\ q \end{pmatrix} = \begin{pmatrix} K_{vc} x_{hd} \\ f \end{pmatrix} \tag{3.140}$$

where $q := [x; \delta] \in \Re^6$, $x_r, x_{hd} \in$SE(2) are the maximal coordinates of the rigid box and the haptic device, $M_r = 0.001 I_{3\times3}$[kg] is the inertia of the rigid body, $K_{rq} := [K_{rq,x}; 0_{3\times3}] \in \Re^{3\times6}$ with $K_{rq,x} = 10^5 I_{3\times3}$ being the spring between the beam end and the rigid box, $K_r := K_{rq,x} + K_{vc} \in \Re^{3\times3}$ with $K_{vc} = 7500 I_{3\times3}$ being the virtual coupling spring, and B_r is the stabilizing damping for the virtual spring. We also set the rotation of x_{hd} to be zero, since Omega 3® is the device of E(3).

We then implement the virtual coupling according to [4], that were derived to be applicable for the PMI while also enforcing (hybrid-time) two-port passivity of the virtual coupling. For instance, with no virtual coupling damping, the force feedback

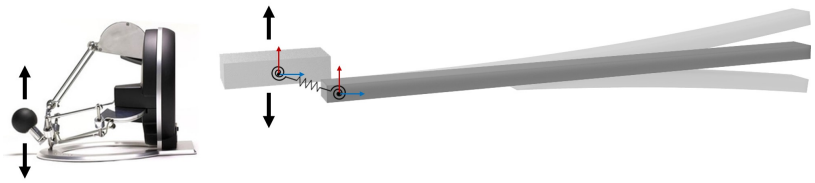


Figure 3-21: Flexible beam (in generalized coordinates) is connected to rigid body (in maximal coordinates) via spring, which is in turn connected to haptic device via virtual coupling.

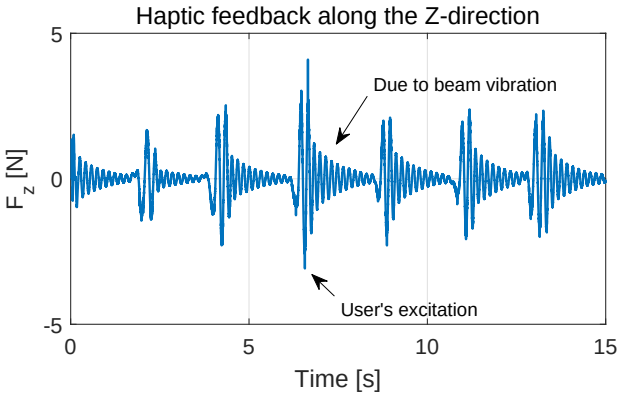


Figure 3-22: Haptic feedback from the flexible beam to the haptic device through virtual coupling connected to the rigid box.

to the haptic device is given by:

$$f_{hd}(t) := -K_{vc}(x_{hd,k} - x_{r,k}), \quad t \in [t_k, t_{k+1}) \tag{3.141}$$

which, even if containing $x_{hd,k}$ and $x_{r,k}$ instead of (continuous-time) $x_{hd}(t)$ and (acausal) $\hat{x}_{r,k}$, can still be passified by the device damping in the continuous-time domain and by the virtual damping B_r in the discrete-time domain. The device damping of our adopted Omega 3® device turns out to be fairly small. This then requires us to choose rather small force scaling factor $\alpha = 0.001$. This small α is in fact an inherent restriction imposed by the device itself (i.e., z-width ([84])). See ([50, 4]) for more details on the derivation and passivity of the virtual coupling for the PMI. Of course, we may choose larger (i.e., sharper) gain K_{vc} if we adopt less conservative techniques than the PD (proportional-derivative) type virtual coupling (e.g., passive set-position modulation [25]).

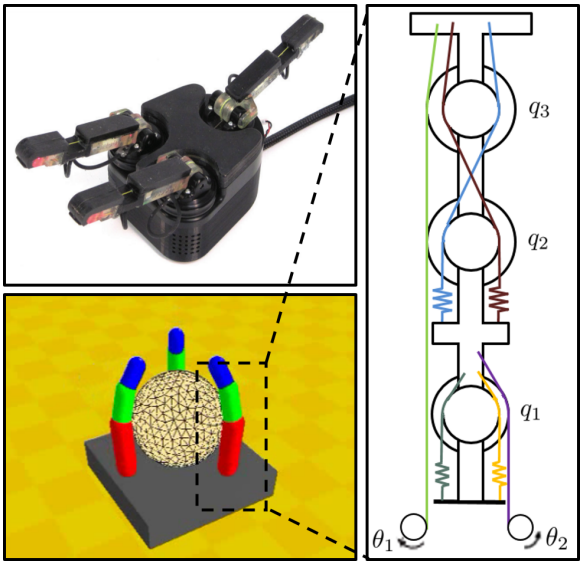


Figure 3-23: The iRobot-Harvard-Yale (iHY) Hand (left), the three fingered virtual UATD hand used in the simulation (middle), and the tendon routing of each finger (right.)

Profile of haptic feedback along the vertical z-direction of the haptic device during the interaction with the flexible beam is shown in Fig. 3-22, where the user haptically interacts with the (lossless/marginally-stable) flexible beam through the rigid box and virtual coupling, some formulated in generalized coordinates and some in maximal coordinates.

3.5.3 Under-Actuated Tendon-Driven Hand Grasping

The last example is the adaptive power-grip of virtual rigid ball by multi-fingered under-actuated tendon-driven (UATD) hand as shown in Fig. 3-23. For the hand, we emulate the configuration of the iRobot-Harvard-Yale (iHY) Hand [85]. We yet model each finger to be constructed according to the design of [86]. For this case, it is then much more convenient to model each finger in generalized coordinates (i.e., joint angles), whereas the hand palm and the virtual ball in maximal coordinates in SE(3).

Each finger is of three-DOF, yet, with only two-DOF actuation via active tendons. See Fig. 3-23 for their tendon-routing (adopted from [86]) with two active tendons

(green and orange lines) and four passive tendons (other lines). The configuration of the UATD hand is then given by the nine joint angles $q \in \Re^9$, which are actuated by the six actuators (i.e., motors) with their angles denoted by $\theta \in \Re^6$. The total stiffness of the three fingers, stemming both from the active and passive tendons, can then be computed more conveniently in the joint angles q (i.e., generalized coordinates) s.t.,

$$K = J_a^T K_a J_a + J_p^T K_p J_p \tag{3.142}$$

where $K_a \in \Re^{6\times 6}$, $K_p \in \Re^{12\times 12}$ are the active and passive tendon stiffness, and $J_a \in \Re^{6\times 9}$, $J_p \in \Re^{12\times 9}$ are their respective Jacobians. Here, K_a, K_p are typically assumed constant (i.e., linear tendon), whereas J_a, J_p are also constant, as they are defined by the pulley radii at each joint of the active and passive tendons, with $J_a q$ and $J_p q$ being their respective length change. The dynamics of the UATD hand can then be written w.r.t. the generalized coordinates q s.t.,

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + Kq = f + f_c \tag{3.143}$$

similar to (2.4), where M, C are the inertia and Coriolis matrices, Kq is the total tendon compliance, f_c is the (external) grasping force, and f is the joint torque control input as given by

$$f = -J_a^T K_a R_a \theta =: -J_\theta \theta \tag{3.144}$$

where $R_a \in \Re^{6\times 6}$ is the matrix of the motor pulley radii (i.e., the tendon tension $K_a R_a \theta$ converted to the joint torque via J_a^T).

To control the adaptive under-actuated power-grip of this UATD hand, we use the Omega 3® haptic device as in Fig. 3-21 of Sec. 3.5.2. Here, we only use one-DOF of the haptic device (i.e., z-direction vertical motion) to command the power-grip. For this, we first map $z \in [z_{\min}, z_{\max}] \in \Re$ of the haptic device to the line interval

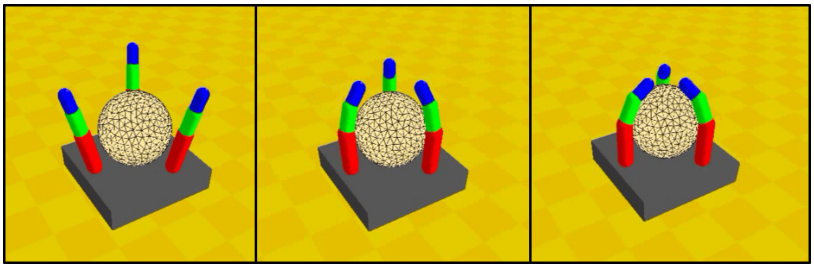


Figure 3-24: Snapshots of adaptive power-grasping of virtual rigid ball by the three-fingered UATD hand, with each finger having tendon-routing as depicted in Fig. 3-23, with the three fingers simultaneously controlled by one-DOF of the haptic device.

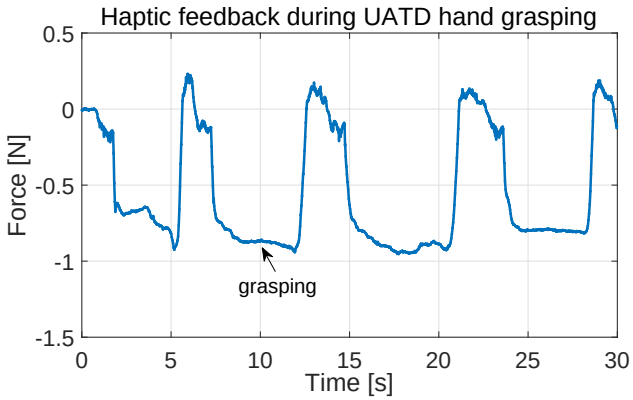


Figure 3-25: Haptic feedback of the one-DOF (i.e., z-directional) contact force of the adaptive power-grasping of the UATD hand in Fig. 3-24.

$q^d \in \text{line}(q_1^d, q_2^d)$ of the UATD hand configuration s.t.,

$$q^d(z) := q_1^d + \frac{z - z_{\min}}{z_{\max} - z_{\min}}(q_2^d - q_1^d) \tag{3.145}$$

where q_1^d, q_2^d represent unfolded and folded configurations. Here, q_2^d is not necessarily required to form the shape of the object to grasp, since the UATD hand can adaptively grip objects with different shape via its tendon compliance and under-actuation. Then, using the quasi-static equation of (3.143), we can compute the following control input θ to attain $q^d(z)$ s.t.,

$$\theta = -(J_\theta^T J_\theta)^{-1} J_\theta^T K q^d(z) \tag{3.146}$$

Of course, the UATD hand in general cannot exactly follow this $q^d(z)$-command,

since its dynamics (3.143) is under-actuated, thus, the contact force f_c can induce (unintended) deformation of the fingers. We provide this information to the user so that s/he can adjust the adaptive grasping in the form of the following haptic device force feedback:

$$f_z(z, z_{\text{proj}}(q)) := -k_{hd}(z - z_{\text{proj}}(q)) \tag{3.147}$$

where $k_{hd} > 0$ is the feedback gain, z is the device current z-position, and $z_{\text{proj}}(q)$ the mapping of the UATD hand configuration q to the z-direction of the haptic device, s.t.,

$$z_{\text{proj}}(q) := z_{\min} + (z_{\max} - z_{\min})\frac{\|q - q_1^d\|}{\|q_2^d - q_1^d\|}. \tag{3.148}$$

Snapshots of this adaptive power-grip of the multi-fingered UATD hand are shown in Fig. 3-24 along with the contact force feedback in Fig. 3-25. For this, we use $K_a = 100I_{6\times6}$[N/m] and $K_p = 10I_{12\times12}$[N/m] with the mass of each link of the finger to be 0.001[kg]. From Fig. 3-24 and Fig. 3-25, we can then see that our PMI-based framework can stably simulate this power-gripping of UATD hand, through the PMI and the multi-point PMI-LCP formulated both for the fingers (i.e., in generalized coordinates: Sec. 3.1) and for the virtual ball and palm (i.e., in maximal coordinates: Sec. 3.2). Note also that the fingers are fairly light/stiff systems, which are difficult to stably simulate by using other techniques (e.g., SEI).

Chapter 4

Contact Clustering and Data-Driven Approach

4.1 Motivation

In this section, we introduce the issue of contact clustering (i.e., clustering contact points and/or contact normals into a manageable number of clusters) which is one of the key process in a real-time contact simulation. The contact simulation typically consists of the following three procedures: (1) contact detection, (2) contact clustering (of contact points/normals) and (3) contact solving. See Fig. 4-1. The problem of contact detection has been more or less settled down with standard approaches available (e.g., [48, 49]). The problem of contact solving, as introduced and developed in previous chapter 3.3, also boasts a large number of well-established techniques with the following two main categories: 1) constraint-based contact solving with linear complementarity problem (LCP) formulation (e.g., [40, 87, 10, 66, 5, 88]; also adopted by ODE [17]) or that with optimization formulation (e.g., [44, 45]; and [46], which renders formulation invertible with some regularization); and 2) penalty-based control solving (e.g., [14, 8, 47]), which is typically simpler/faster than the constraint-based methods with the price of inter-object penetration.

However, the problem of contact clustering is much less investigated than the other two procedures. This contact clustering is imperative to achieve reasonably-fast speed

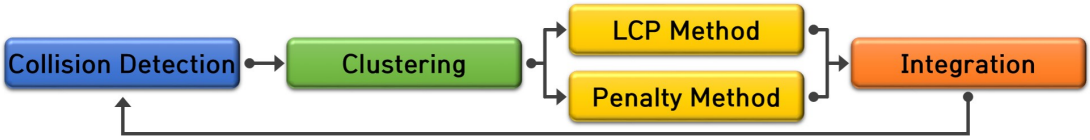


Figure 4-1: Simulation process with multi-point contacts. Contact clustering is employed for real-time fast and numerically stable contact simulation.

of the simulation calculation while also enhancing numerical stability of the contact simulation. For instance, large number of contact points can be produced when fine mesh models are used or the contact geometry is complex. While typically making the contact overdetermined, large number of contact points produce ill-conditioned contact matrix of (3.79) especially when friction is add. Solving this ill-conditioned contact is difficult and sometimes fails to find solution. It can also slow down the convergence of iterative solvers. Even without the ill-conditioning issues, large number of contact points take more time to compute contact force. Therefore, contact clustering is typically employed to reduce the number of contact for real-time simulation.

Some of rare, yet, to our knowledge, still representative works in this area are [14, 88], both standing upon the k-means clustering technique [89]. The work of [14] proposed a k-means contact clustering based on contact positions and penetration depths. This clustering is then used for the penalty-based contact force calculation to enhance interaction stability by reducing the number of contact points, which consequently limits the maximum contact stiffness. The work of [88] presented a k-means contact clustering in the seven-dimensional space of contact positions, contact normals and friction coefficients, the result of which is then used for the LCP-based contact force calculation. They also reported that their seven-dimensional space clustering garners more stable contact than, e.g., the simple contact sorting of Gazebo [90]. As can be seen from these, the k-means clustering is rather the state-of-the-art for the contact clustering, which is also adopted by Bullet simulator [19] although its details not fully disclosed. Even so, none of these works explicitly establish the accuracy of their scheme against the real experimental data.

Along this line, in this section, we propose a novel data-driven learning-based contact clustering framework for rigid-body simulation with its accuracy established/ver-

ified by using real experimental data. Here, we believe the data-driven/learning-based approach is suitable, since the mapping from the contact clustering to the contact simulation accuracy is, in general, difficult (or, more often, impossible) to derive analytically. This is also motivated by our experience that, by hand-tuning the contact clustering, sometimes, the contact simulation results seem more plausible. For this, we first construct an experimental robotic setup with force/torque (F/T) sensors to collect real contact motion/force data. We then design a multilayer perceptron (MLP) network for the contact clustering, which utilizes the full motion (i.e., relative position and velocity) and force/torque information of the colliding rigid body. It is worthwhile to mention that this full motion/force/torque information is typically not used in the works of the contact clustering based on the k-means clustering (e.g., [14, 88]), although the contact behavior would likely depend on this full motion and force/torque data during the contact. For our framework, we also adopt the constraint-based optimization contact solver similar to [46], since, from its being invertible, it facilitates the learning of our MLP-network during the training. Our proposed data-driven/learning-based contact clustering framework is then verified against the experimental setup along with other techniques/simulators and shown to significantly enhance the accuracy of contact simulation.

4.1.1 K-Means Clustering

Here, we briefly introduce the k-means clustering which is often employed for general mesh-mesh contact [88, 14]. The k-means clustering simply produces k clusters from l_c observations (contact points or normals) where the cluster number k is predefined by user. It consists of three steps:

- Initialize: Choose k observations randomly and use them as means of each cluster.

- Assignment: Assign each observations to the nearest cluster. The euclidean distance between the mean and the observation is used.

- Update: Calculate the new means of the new cluster. Go to the assignment step until the clusters do not change.

This algorithm is simple but has an issue that it can produce different clusters depending on the initial guess. For a continuing contact, the change of resulting clustered contact points can produce chattering in calculated contact forces. In this case, to guarantee temporal coherence, *warm starting* can be employed where the previous step's clustering is used for the initial guess of the next clustering of continuing contact.

Our prime motivation is that, this clustering affects the accuracy of simulation, not only the stability. However, to our best knowledge, there is no research which analyses the effect of this clustering on the accuracy of contact simulation, since its analytical formulation is in most cases impossible. For this, in the next section, we propose a novel data-driven contact clustering for the contact simulation with significant improvement of simulation accuracy as verified by real-world experiments.

4.2 Data-Driven Contact Clustering

In this section, we propose our data-driven contact clustering which consists of two parts (See Fig. 4-2 for its structure). The first part is the multilayer perceptron (MLP) network $fn(X_k)$ [91] to classify which contact is on or off. The input of this MLP network is the dynamics-related information $X_k := [x_k; R_k; v_k; \omega_k; f_k^e; \tau_k^e] \in \Re^{24}$ which is the position, orientation with stacking each column of R_k, linear/angular velocity, and external force and moment in order, where the angular velocity and external moment are defined in its body frame following the formulation of our PMI in Chap. 3 and Chap. 3.3. Whereas general contact clustering methods only consider the geometric information such as contact points, normals, or penetration depths [88, 14], here we employ this dynamics-related input X_k, since it is indeed physically reasonable to infer that the clustered contact changes depending on the dynamic state. This input X_k is then processed by one hidden layer with its size being 40. Here, our experience is that the depth of the network should not be so deep (the

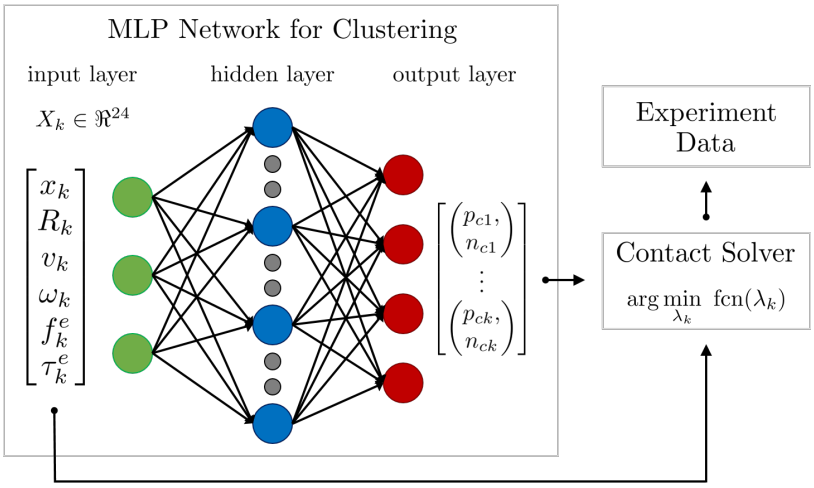


Figure 4-2: Architecture of our data-driven contact clustering: MLP network for clustering and constraint-based contact solver. The input of the MLP network is dynamics information of $\Re^{24}$ with single depth hidden layer.

network with one or two hidden layers works well). Over-fitting problem arises with 4 or 5 hidden layers. After this, the output classification layer indicates whether some of the given contact points and/or normals are selected as an active contact or not. Consequently, the input of the MLP network is $\Re^{24}$ vector and the output is $\Re^{n_c}$ vector with its component meaning whether the contact is activated or not from a given contact grid.

The second part of our data-driven contact clustering is the constraint-based contact solver. This constraint solver receives the output of the MLP network, contact points and normals, with the dynamics information and computes the contact force. For this, we reformulate (3.79)- (3.80) replacing tangential velocity part with (3.96) as an optimization problem while introducing a force minimization following [46]:

$$\arg\min_{\lambda_k} \quad \sigma_k^T \bar{H}_k^{-1} \sigma_k + \lambda_k^T R \lambda_k \tag{4.1}$$

$$\text{subj.} \quad \sigma_k^n \geq 0, \ \lambda_k^n \geq 0, \ \mu|\lambda_k^n| \geq |\lambda_k^t| \tag{4.2}$$

where $\sigma_k := [\sigma_k^n; \sigma_k^t] \in \Re^{3l_c}$ are the normal and tangential velocity, $\lambda_k := [\lambda_k^n; \lambda_k^t] \in \Re^{3l_c}$ are contact force along normal and tangential direction, $\bar{H}_k \in \Re^{3l_c \times 3l_c}$ is the

contact matrix with

$$\bar{H}_k := \begin{bmatrix} \bar{N}_k^T A_k^{-1} \bar{N}_k & \bar{N}_k^T A_k^{-1} \bar{D}_k \\ \bar{D}_k^T A_k^{-1} \bar{N}_k & \bar{D}_k^T A_k^{-1} \bar{D}_k \end{bmatrix} \tag{4.3}$$

constructed by using (3.79)-(3.80) and (3.96), and $R := \text{diag}[r_1, r_2, \cdots] \in \Re^{3l_c \times 3l_c}$ is the positive regularizing matrix. The physical meaning of (4.1) is that, it maximizes the dissipation due to the contact while also minimizing the norm of total contact forces by R. Then, the solution of (4.1) can be obtained by solving

$$(\bar{H}_k + R)\lambda_k + \gamma_k = 0 \tag{4.4}$$

where $\gamma_k := [\gamma_k^n; \gamma_k^t]$, with a PGS-like iterative method under the constraint (4.2), which projects the nonlinear Coulomb friction similar to that of [13]. Note that every parameters of this optimization can be calculated from the network input X_k and inertial parameters (i.e., m or J).

4.3 Data Acquisition and Network Training

Now, we train the MLP network $fn(X_k)$ with a standard cross-entropy loss function. For this, a desired active contact points and normals should first be given. This desired active contact points and normals are calculated by solving (4.1)-(4.2) and matching its result with the experiment data. For this, we configure the experiment environment as shown in Fig. 4-3. The experiment environment consists of 7-DOF robot manipulator (KUKA LBR iiwa 14 R820), two ATI Gamma F/T sensors, and aluminium peg and (static) plate. The mass and inertia of aluminium peg are $m = 339.53$ [g] and $J = \text{diag}[182767.90, 182767.90, 106769.63]$ [g×mm^2] where its center of mass is 34.54 [mm] away from the F/T sensor along the z-direction in upright posture. The two F/T sensors are attached to the robot end-effector and the ground to measure the external force acting on the peg and the contact force between the peg and the plate. Then the torque controlled robot manipulator make the peg

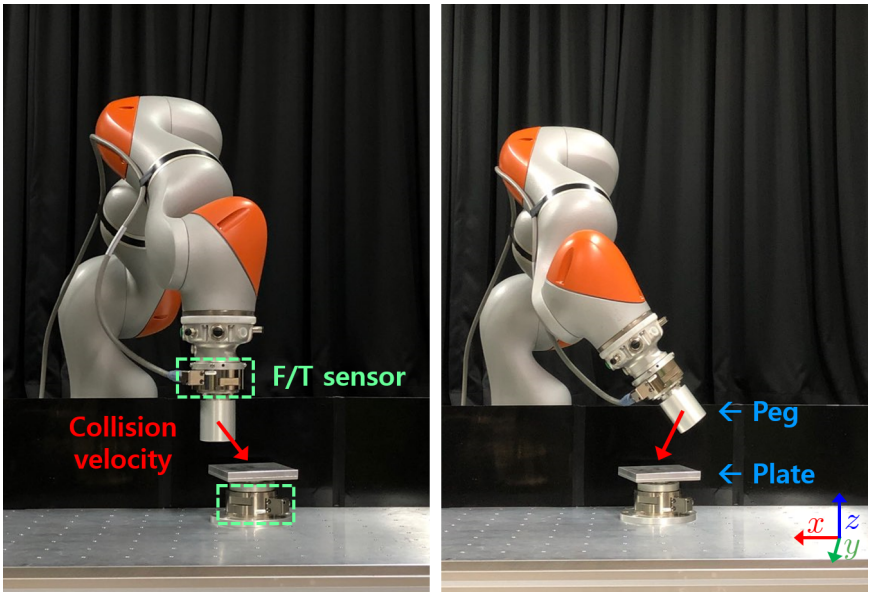
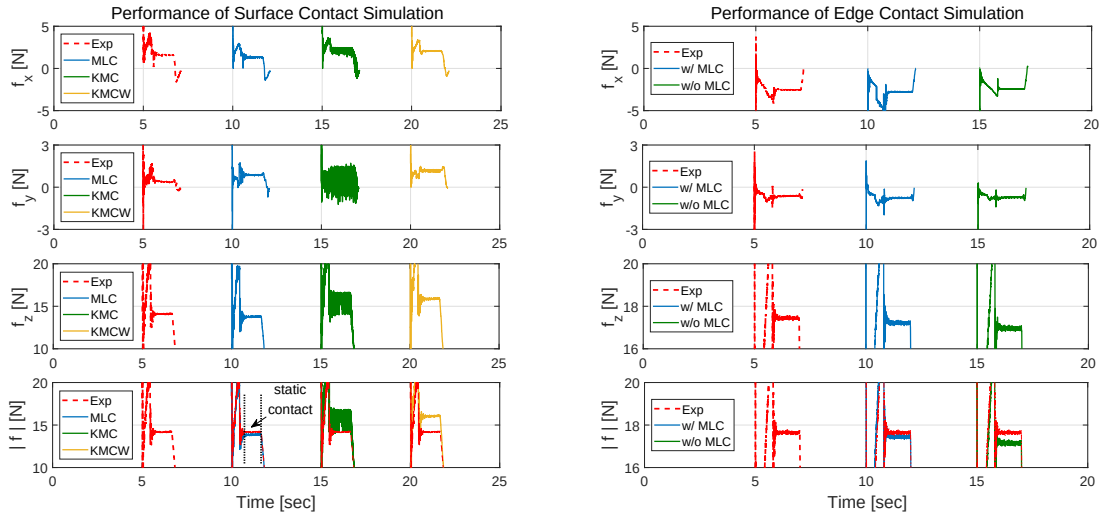


Figure 4-3: Experiment setup consists of one KUKA robot manipulator, two ATI gamma sensors, and aluminium peg and plate. Vertically placed peg collides to the plate with different velocity for the first contact scenario (left) and inclined peg collides with different velocity for the second contact scenario (right).

colliding with the plate. To make the peg contact with the plate along the predefined path, we design the end-effector impedance control [53] with the control gains $K = [800, 800, 800, 150, 400, 50]$ and $B = [250, 250, 250, 15, 30, 15]$ for each $\{x, y, z\}$ translation ([N/m] or [Ns/m]) and rotation ([Nm/rad] or [Nms/rad]).

For a contact data acquisition, we design two contact scenarios. In the first contact scenario, we make the peg contact to the plate with its surface to produce multiple contact points (left of Fig. 4-3). To obtain various contact data for this surface contact geometry, we change the approaching velocity, i.e., the incidence angle from 45° to 135° with 10 uniform grids and the magnitude from 0.02 to 0.07 [m/s] with 10 uniform grids. Thus, we obtain contact data for one hundred contact cases in this scenario. In the second scenario, we make the peg contact with its corner to the plate (right of Fig. 4-3). In this case, we change not only the collision velocity but also the attitude of the peg from 135° to 175° rotated along the y-axis. Since we used 5 grid points for this second scenario, total 125 contact data was obtained (i.e., 5 for incidence angle, 5 for magnitude, and 5 for peg attitude). In Sec. 4.4, we will present the performance of our proposed data-driven contact clustering and some



(a) Surface contact evaluation

(b) Edge contact evaluation

Figure 4-4: Contact simulation performance: contact force of (4.1)-(4.4). For the surface contact (left), contact force with the machine learning-based clustering (w/ MLC, blue), with the k-means clustering (w/ KMC, green), and with the k-means clustering and warm starting (w/ KMCW, yellow) are presented. For the edge contact (right), contact force with the machine learning-based clustering (w/ MLC, blue), and without machine learning-based clustering (w/o MLC, green) are presented.

representative open-source robot simulators.

4.4 Performance Evaluation of Data-Driven Clustering

In this section, we present the performance of the data-driven contact clustering proposed in this Chap. 4. We plot the resulting force of (4.1)-(4.4) overlaid with the measured contact force from the experimental setup. The input X_k is directly calculated by using manipulator's joint angle and velocity, and the F/T sensor attached to the peg. We manually measure the friction coefficient $\mu = 0.15$ while the restitution coefficient $\epsilon = 0.01$ being empirically estimated to make the contact behavior as close as possible to the experiment data. We also compare the contact force for the same input X_k obtained by (4.1)-(4.4) with the standard k-means clustering and the k-means clustering with warm starting.

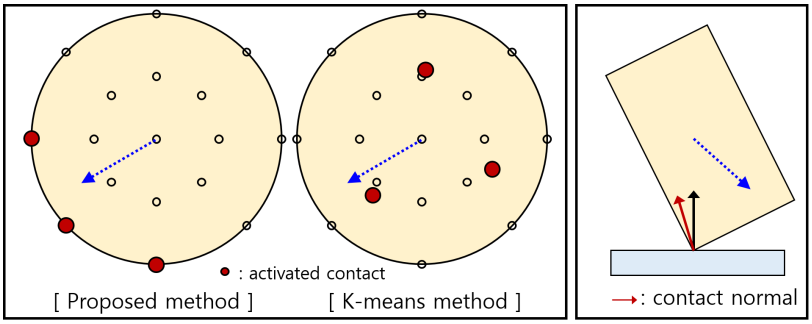


Figure 4-5: Illustrative example of learning-based clustering: learning-based clustering produces different contact points compared to standard k-means clustering (left) or normal (right) depending on X_k (e.g, external force and colliding velocity).

For the first surface contact scenario, we take 17 contact points as the output of the MLP network $fn(X_k)$. In this case, since both contact spaces of the peg and the plate are plain, we take all the normal of each contact point as z-direction (i.e, $n_k^i = [0; 0; 1]$). We also take three contact points as the output contact points since three points is the minimum number of points to define surface. Then, we find optimal contact point pairs by solving the calculated force of (4.1)-(4.4) and comparing it with the experiment data. We use 70% of total contact data to train the network and use the rest 30% data to test the resulting contact force.

The performance of each contact simulation for surface contact scenario is shown in Fig. 4-4(a). For readability, we shift the starting time of each force data as 5 seconds and we overlay the experiment data on the magnitude plot of each simulation (fourth plot) for more better comparison. It is shown that, with the proposed data-driven contact clustering, the accuracy of the calculated contact force increases. In particular, the normal direction force (third plot) is the same as the experiment. On the other hand, with the only standard k-means clustering, there is errors not only in the friction, but also in the normal direction. In addition, it is shown that the output contact force significantly vibrates if warm starting is not used. After all, the root-mean-square errors (RMSEs) of $(|f_k| - |f_k^{exp}|)/|f_k^{exp}|$ on the static contact of these three cases (i.e., MLC, KMC, KMCW) are 2.3%, 13.8%, and 13.1% respectively. Here we compare the static contact since its accurate generation is the very basic requirement for any simulators and also can be properly measured by the F/T sensors with limited

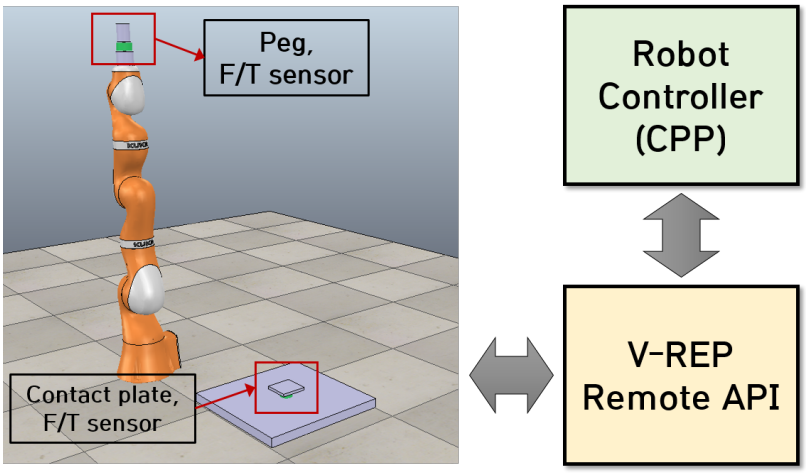


Figure 4-6: Configuration of V-REP robot simulation. Same real KUKA robot controller is used to control the virtual KUKA robot via remote API.

bandwidth (against impacts, noise, etc.). In fact, it is truly hard to accurately measure the impact data with F/T sensor, yet the tendency is similar with the dynamic contact (about 11% RMSE for the proposed method and about 24% RMSE for the k-means method).

In Fig. 4-5, we illustrate contact points clustered by the network $fn(X_k)$ and by the k-means clustering. It turns out that the network choose different contact points depending on the colliding velocity (blue arrow in Fig. 4-5). This shows that our proposed clustering method, which fully exploit the dynamics-related information (e.g., velocity and external force etc.), can significantly increase the accuracy of multi-point contact simulation. Note that, we can expect the accuracy to be increased by employing normal direction clustering, yet, we do not use it in this surface contact scenario since 1) the normal contact direction is almost z-direction due to its geometry and 2) it will reduce the optimal points searching time.

For the second edge contact scenario, we fix the contact point since it will produce almost single contact point between edge of cylinder and plain. However, we divide contact normals into seven grids of $\pm 45°$ from the z-axis along the x- and y-axes (consequently 49 grid normals). Although it is easy to take $n_k^i = [0; 0; 1]$ from the plate, there exists unmodeled contact geometry such as chamfer, filleted-corner, or rough of surface etc. This normal grid can reflect such unmodeled geometry for the

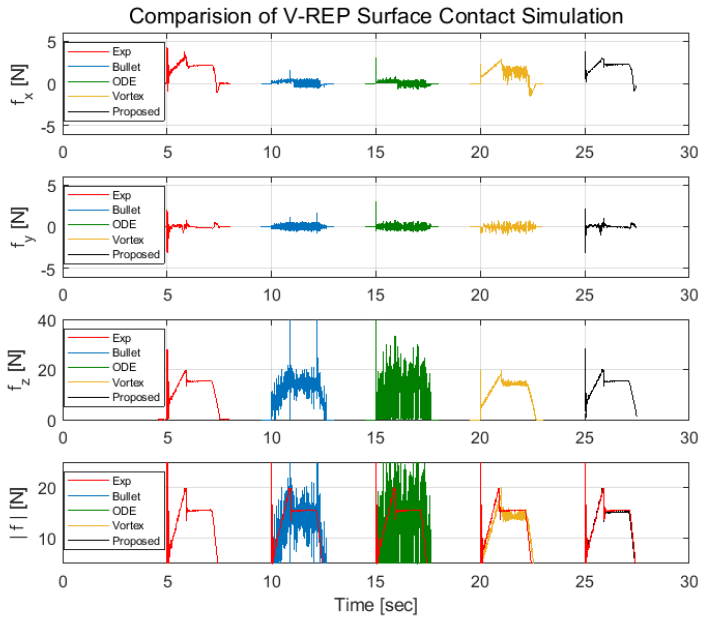


Figure 4-7: Comparison of surface contact simulations done by V-REP and experiment. We also plot the result of our proposed data-driven contact result.

contact solver (4.1)-(4.4). In Fig. 4-4(b), we present the resulting contact force. As a comparison, we also plot the contact force calculated with $n_k^i = [0; 0; 1]$ (simply obtained from the geometry) for the same contact point. The results show that the MLP network actives $n_k^i = [0; 0; 1]$ as the contact normals in many cases, but there are still many other normal direction which shows better contact performance. See Fig. 4-5 left for the illustrative contact normal activation during edge contact. For this contact, the RMSEs with and without the learning-based clustering are 2.0% and 2.9% respectively. Although both contact simulations shows quite accurate performance for this case, it is obvious that the proposed method can increase the accuracy of contact simulation.

4.4.1 Comparison with Open-Source Robot Simulators

Here we illustrate performance comparison of existing open-source robot simulator: Bullet, ODE, and Vortex. Each physics engine is widely used for robot simulation (e.g., bullet for Roboschool with OpenAI Gym, ODE for CHAI3D, and Vortex for

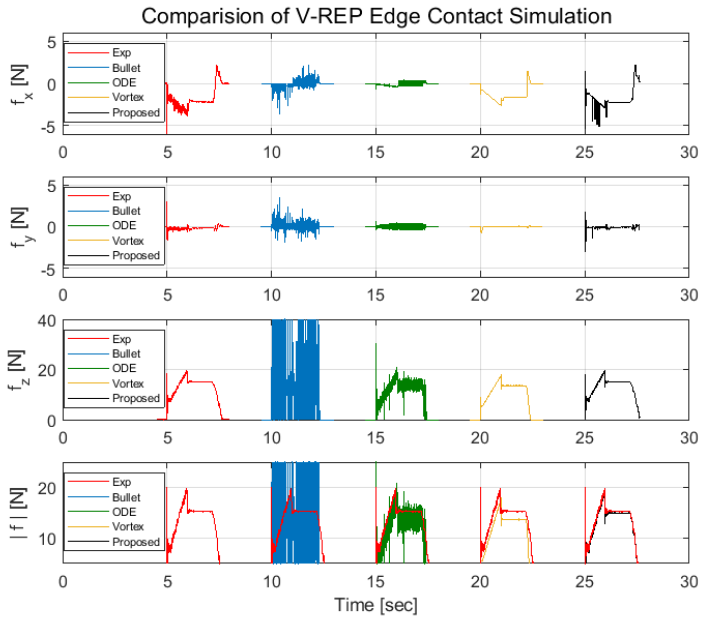


Figure 4-8: Comparison of edge contact simulations done by V-REP and experiment. Black line represents the result of our proposed data-driven learning-based contact.

Case	Bullet	ODE	Vortex	Proposed
Surface contact	10.4%	27.6%	7.8%	2.3%
Edge contact	78.5%	10.7%	6.7%	2.0%

Table 4.1: Performance evaluation and comparison of V-REP simulators and the proposed data-driven contact clustering.

many industrial training simulators). To compare them in identical environment, we use V-REP robot simulation platform [92] with its remote API to control the virtual robot with CPP controller which is used for the real KUKA robot. The configuration of V-REP simulation with the remote API is illustrated in Fig. 4-6.

To make both virtual and real KUKA robot as same as possible, we fine-tune kinematic parameters (e.g., joint length and its relative position) and dynamics parameters (e.g., mass, inertia, and position of center of mass) of simulators. For the dynamics parameters, we use the identification of [93]. On the other hand, it is hard to stably simulate stiff system (e.g., large K gains) in real-time unless using special simulation such as PMI proposed in this thesis. Indeed, if we use the same controllers for the real-time simulation of V-REP, it becomes unstable. Therefore, to obtain

stable simulation with the V-REP, we run the V-REP simulation in non-real-time manner with $T_k = 0.2$ [ms] step size. Note that, this $T_k = 0.2$[ms] is the maximum step size that can stably implement the controller used in the experiment. Compared this V-REP, the PMI simulation maintains stability under the varying step size. With the PMI simulation, for a similar 7-DOF robot simulation with contact, the step size varies from $T_k = 0.0006$[ms] to $T_k = 0.0047$[ms] while running the simulation in real-time. This means that, the PMI simulation can stably simulate with at least 23.5 times larger time step compared to the V-REP for such robot contact simulation. In fact, for a single contact task scenario (e.g., peg-in-hole) which takes about 28 sec in real-time, the V-REP simulation takes about 767 sec which means that, PMI simulation is at least 27.39 times faster than the V-REP simulation when implementing such stiff system with large control gain. Note also that we do not use Newton simulator which is also provided in the V-REP since it becomes unstable with the given configuration. In Fig. 4-7 and Fig. 4-8, we plot contact force data of the surface and the edge contact scenarios respectively, in order of experiment, Bullet simulator, ODE simulator, and Vortex simulator. As in Sec. 4.4, we shift the starting time of each force data as 5 seconds for readability. Although there is some vibration, Vortex simulator shows the best performance among V-REP simulators. Other Bullet and ODE simulators exhibit significant chattering in the force data even though their behaviors are similar to the naked eye.

Finally, we summarise RMSEs during the static contact of V-REP simulators and the proposed method in Table. 4.1. The Vortex simulator shows the best performance among V-REP simulator (7.8% and 6.7% RMSE for both scenarios). However, our proposed data-driven contact clustering can produce more accurate contact force than Vortex simulator.

Chapter 5

Application to Multiuser Haptic Interaction with Wearable Haptic Interface

As an illustrative application of the PMI-based haptic rendering, here, we present the hand-based collaborative manipulation of a shared rigid body among two user through the network communication. Two technical issues in constructing virtual haptic interaction over the network communication are: 1) consistency and 2) the communication delay. The consistency means that the virtual haptic scene displayed to each geographically distributed remote user should be identical as much as possible. Thus, this consistency is related to the performance of the haptic interaction over network. On the other hand, the communication delay can easily destabilize the interaction system when there is haptic feedback loop. For example, assume that an user tries to touch a remote wall. If the wall impedance is increased since he/she want to render a stiff virtual wall, it suddenly starts vibrating with some level of wall stiffness. In addition, the renderable level of wall stiffness decreases with the increasing of communication delay. So, this communication delay is related to both stability and performance. In the following, we present the construction of networked haptic interaction system among two remote users. We also show that it is possible to realize a stable networked haptic interaction, by exploiting the discrete-time passivity

of the PMI simulation. The experimental result clearly shows that the unstable behavior occurs when we use non-passive simulation (e.g., Bullet) which implies that the passivity of PMI simulation plays a crucial role for the delay stabilization.

5.1 Design of Peer-to-Peer Control Architecture with Consensus Control

There are two ways to design the control architecture of remote haptic interaction: 1) client-serve control architecture ([94, 95]) and 2) peer-to-peer control architecture ([30, 96]). In the client-server control architecture, the simulation of virtual scene and the haptic rendering are done in the server and each client PC measures/sends each user's motion information and displays haptic feedback. This architecture has a benefit especially in the consistency, since each user receives a haptic and graphic information from a single virtual world of the server. However, since the motion information of local user should first be sent to server and then receive feedback information from the server, there is a lag between motion and its feedback (including graphics and haptics). In addition, the information of remote user will arrive at local site passing through the server, it has the effect of increasing the communication delay.

On the other hand, in the peer-to-peer control architecture, each user simulates their own virtual world and the consistency is obtained via the so-called consensus control[30, 97]. In this architecture, unlike the client-server control architecture, the feedback corresponding to a local user's motion is displayed instantaneously to the user since each virtual world and feedback is calculated in local site. In addition, since the information of the remote site is directly sent without passing the server, there is no delay increase. However, since each user has their own virtual world, this architecture, the design of the consensus control is crucial for performance and stability of whole networked haptic interaction system.

As an illustrative application of the PMI-based haptic rendering, in this thesis,

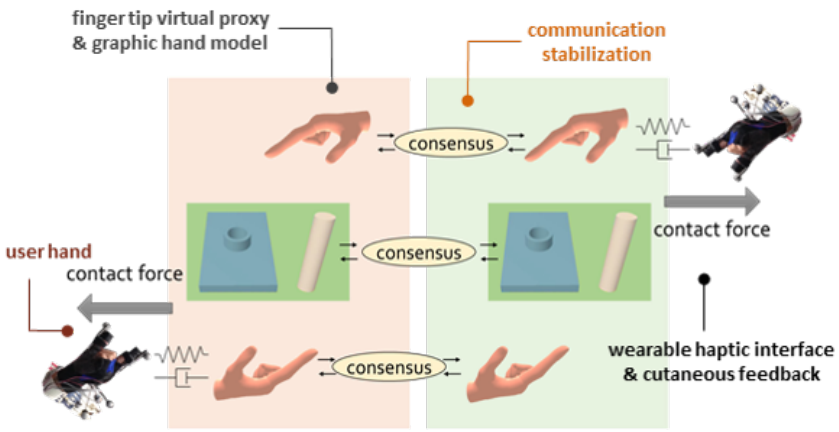


Figure 5-1: Peer-to-peer control architecture for the collaborative peg-in-hole haptic interaction among two remote users.

we develop haptic enabled multiuser collaborative rigid-body peg-in-hole assembly among two remote users based on the peer-to-peer architecture. The overall system configuration is shown in Fig. 5-1. Each local virtual world consists of multi-fingered hand virtual proxies of local user and remote user, virtual peg, and virtual hole. The hand virtual proxies and virtual objects are then implemented by using the rigid body formulation as introduced in Sec. 3.2. So, for the consistency, by using $d\psi_k^T$ in (3.44) with a local damping injection, we can design the SE(3) consensus control for ith rigid body as

$$ {}^{p}\tau_{i,k}^{t} := -B^{tp}\hat{v}_{i,k} - K^{t}({}^{p}\hat{x}_{i,k} - {}^{q}\bar{x}_{i,k}) \tag{5.1} $$

$$ {}^{p}\tau_{i,k}^{r} := -B^{rp}\hat{\omega}_{i,k} + K^{r}\frac{{}^{pq}\phi_{i,k}}{2\sin{}^{pq}\phi_{i,k}}({}^{p}R_{i,k}^{T}\,{}^{q}\bar{R}_{i,k} - {}^{q}\bar{R}_{i,k}^{T}\,{}^{p}R_{i,k})^{\vee} \tag{5.2} $$

where $B^t, B^r, K^r, K^r \in \Re$ are the local damping injection and the proportional consensus gains for each translational and rotational dynamics, ${}^{pq}\phi_{i,k} = \cos^{-1}\frac{1}{2}(\mathrm{Tr}({}^{p}R_{i,k}^{T}\,{}^{q}\bar{R}_{i,k}) - 1)$ is the rotational distance used in (3.35), and ${}^{q}\bar{x}_{i,k}$ and ${}^{q}\bar{R}_{i,k}$ are the set-position and the set-rotation of the ith rigid body of remote site q which is corresponding to the ith rigid body in local site p. Note that, the $\vee$ is inverse skew-symmetric operator.

Now, the easiest way to implement (5.1)-(5.2) is just using the position and rota-

tion information of remote copies ${}^q\hat{x}_{i,k-N_k}$ and ${}^qR_{i,k-N_k}$ which arrives at kth simulation step (note that here we assume that the information delay of local to remote and remote to local communication are same as N_k for simplicity of notation). In this case, it is possible to passify (5.1) with a damping injection amount to

$$B^t \geq N_k T_k^{\max} K^t \tag{5.3}$$

where $T_k^{\max} := \max(T_k)$ as shown in [30]. However, it is hard to analyse the appropriate damping injection B^r for the rotational consensus control (5.2) since it is nonlinear SO(3) spring. One can expect that the same damping injection (5.3) works for (5.2) and, indeed, it works practically during the implementation. However, to thoroughly passifying (5.1)-(5.2), we can extend the passive-set-position-modulation (PSPM [25]) for the SE(3) consensus control.

The PSPM [25] is originally developed to prevent sudden potential energy jump from the 'discretization' of set-position signal due to the internet communication. The basic idea of the PSPM is to find a set-position which is closest to the received set-position signal while preventing it from violating the energetic passivity. We can extend this PSPM for the SE(3) rigid body consensus control by considering the translation energy and the rotation energy separately. First, we define the set-position modulation strategy as

$$\underset{{}^q\bar{x}_{i,k}}{\text{argmin}} \quad \|{}^qx_{i,k-N_k} - {}^q\bar{x}_{i,k}\| \tag{5.4}$$

$$\text{subj.} \quad E_{i,k}^t = E_{i,k-1}^t + E_{i,k}^{t,in} + D_{i,k-1}^t - \Delta P_{i,k}^t \geq 0 \tag{5.5}$$

and the set-rotation modulation strategy as

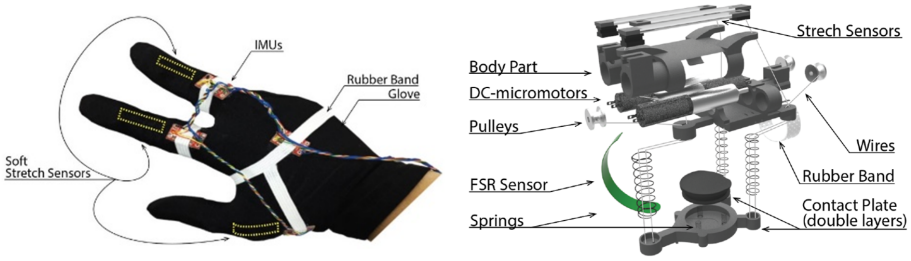
$$\underset{{}^q\bar{R}_k}{\text{argmin}} \quad \phi({}^qR_{i,k-N_k}, {}^q\bar{R}_{i,k}) := \|\log({}^qR_{i,k-N_k}^T {}^q\bar{R}_{i,k})\| \tag{5.6}$$

$$\text{subj.} \quad E_{i,k}^r = E_{i,k-1}^r + E_{i,k}^{r,in} + D_{i,k-1}^r - \Delta P_{i,k}^r \geq 0 \tag{5.7}$$

where $E_{i,k}^t \in \Re$ is virtual energy reservoir for the translation of ith rigid body,

$D^t_{i,k-1}$ is the translational damping dissipation numerically calculated from the PMI-simulation, and $\Delta P^t_{i,k} := \frac{1}{2}K^t\|{}^p x_{i,k} - {}^q\bar{x}_{i,k}\| - \frac{1}{2}K^t\|{}^p x_{i,k} - {}^q\bar{x}_{i,k-1}\|$ is the possibly passivity-breaking energy jumping due to the set-position change ${}^q\bar{x}_{i,k-1} \to {}^q\bar{x}_{i,k}$. For the set-rotation modulation, the $E^r_{i,k} \in \Re$ is virtual energy reservoir for the rotation of ith rigid body, $D^r_{i,k-1}$ is the rotation damping dissipation numerically calculated same as $D^t_{i,k-1}$, and $\Delta P^r_{i,k} := \frac{1}{2}K^r\phi({}^p R_{i,k}, {}^q\bar{R}_{i,k})^2 - \frac{1}{2}K^r\phi({}^p R_{i,k}, {}^q\bar{R}_{i,k-1})^2$ is the possibly passivity-breaking energy jumping due to the set-rotation change ${}^q\bar{R}_{i,k-1} \to {}^q\bar{R}_{i,k}$. The $E^{t,in}_{i,k}$ and $E^{r,in}_{i,k}$ are the energy input by, e.g., the shuffling/ceiling from the remote site which will be calculated from the excess of $E^t_{i,k}, E^r_{i,k}$ from the maximum energy $\bar{E}^t_i, \bar{E}^r_i$. Note that, as mentioned above, the physical meaning of (5.4)-(5.7) is that, for (5.1)-(5.2), we modulate and use the set-position ${}^q\bar{x}_{i,k}$ and the set-rotation ${}^q\bar{R}_k$ which are close to the raw set-position and set-rotation of ${}^q x_{i,k-N_k}$ and ${}^q R_{i,k-N_k}$ as possible while maintaining energetic passivity (5.5) and (5.7) (i.e., the extent permissible by the available energy). Therefore, we can always passify the consensus control regardless of the communication imperfectness (i.e., delay or packet loss) while maximizing the performance (i.e., maximum information recovery).

Meanwhile, in (5.5) and (5.7), the terms $D^t_{i,k-1}, D^r_{i,k-1}$ reharvest the energy dissipated by the damping and, during simulation, we can exactly capture this damping dissipation. Then, let us assume that there is no other energy dissipation or energy generation. Then, for a constant desired set-position and set-rotation signals, the position and rotation of the rigid body will converge to the desired set-position and set-rotation if time goes by. In addition, during the conversing, all the dissipated energy will be stored to the energy reservoir and it will exactly amount to the initial energy. Since this stored energy will be used for the control when other set-position and set-rotation signals arrive, it means that the possible work done by this rigid body is bounded by the initial energy unless there is no energy injection, e.g., by $E^{t,in}_{i,k}$ and $E^{r,in}_{i,k}$. On the other hand, assume that there is some energy generation due to, e.g., non-passive integration and again we can reharvest all the damping dissipation. Then, the energy stored in the reservoir after the convergence will increase (here we assume that the maximum energy is large enough). This means that even



Finger Tracking Module Cutanous Haptic Module

Figure 5-2: Wearable cutaneous haptic interface consists of two modules, namely: finger tracking module (FTM) for multi-DOF finger/hand motion estimation and cutaneous haptic module (CHM) foe 3-DOF contact force feedback on the fingertips.

if we use this PSPM which can passify the communication, the networked haptic interaction system can be unstable without passive simulation. The importance of passive simulation will be clearly shown in the experiment result of next section.

On the other hand, as shown in Fig. 5-2, we develop and use the wearable cutaneous haptic interface (WCHI) for the multi-DOF finger/hand motion tracking and the three-DOF fingertip contact force feedback [80]. This WCHI is developed since there is no appropriate hand tracking device with simultaneous haptic feedback. Here the motion of hand is applied to the hand virtual proxies through the SE(3) virtual coupling in Sec. 3.2 while the contact force of the hand virtual proxies are feedback to user. Note that, this WCHI consists of the finger tracking module (FTM) for the finger/hand motion estimation and the cutaneous haptic module (CHM) for the fingertip cutaneous haptic feedback. The uniqueness of FTM is employment of heterogeneous sensors (i.e., inertial measurement units and soft sensors) for the efficient and effective estimation of multi-DOF finger motion within a small/limited area while avoiding electromagnetic interference from an actuator of haptic device or occlusion of vision-based tracking system. Then, the motion tracking of this FTM is achieved by using the forward kinematics, so only the wrist position is required, which is relatively easy to estimate from outer vision sensor (e.g., VIVE tracker or MOCAP) without occlusion. On the other hand, we employ the cutaneous haptic feedback which displays contact force feedback by using skin deformation. This cutaneous haptic feedback is especially appropriate for the hand, since it can generate

three-DOF contact force without bulky and complex mechanism of kinesthetic haptic (thus, minimizing motion disturbance). By constructing direct feedback control with one FSR sensor and three soft sensor, this CHM achieves robust the haptic rendering performance across different user. We also want to note here that this cutaneous haptic feedback is local force feedback, thus does not affect the finger motion. This decoupling, as mentioned in [82], make the stability of the networked haptic interaction system free from the haptic rendering. Therefore, it is more clear to see the relation between passive simulation, delay passivation, and stability of networked haptic interaction.

5.2 Experimental Results and Performance Comparison

For the implementation of the networked haptic interaction system with the peer-to-peer control architecture, we use two PCs and one internet router. Through this router the two PCs are directly connected via Ethernet communication constructing local area network (LAN). Then, the two PCs communicate with each other through UDP communication which is faster than the TCP/IP communication. Here note that we send and receive the position and the rotation information rather than the velocity of wave variable (e.g., [96]) it is robust under packet loss which can be arise with UDP communication. Now, since the objective of this application is to validate the robustness of our passive simulation and the passive consensus control under communication delay, we emulate the delay with employing a packet buffer which catches the received packet for some time interval while dropping it with some predefined rate. By doing so, it is possible to implement large time delay with packet loss with the LAN with small delay.

For the virtual world of the haptic collaborative manipulation shown in Fig. 5-3, we set the mass/inertia of rigid bodies of the hand virtual proxies as $m_i = 0.001$[kg] and $J_i = m_i I_{3\times3}$[kgm^2], while the stiffness of the spring stiffness as $k_\lambda = 100$[N/cm],

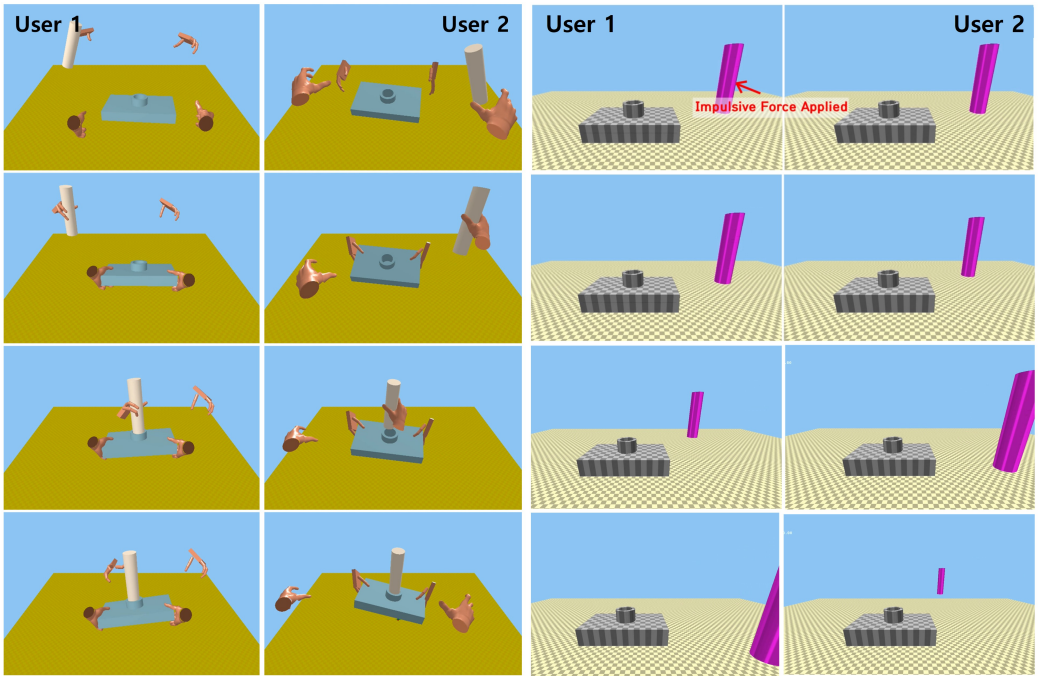


Figure 5-3: Snapshot of collaborative shared virtual rigid body manipulation among two remote users implemented via the PMI simulation (left) and the open-source physics engine Bullet (right). The PMI-based interaction system can maintain its stability with a communication delay larger than 100[ms] under $K_c = 100$[N/cm]. On the other hand, the Bullet-based interaction system becomes unstable with a communication delay larger than 50[ms] under $K_c = 75$[N/cm].

with the stiffness/mass ratio to be fairly high of 10^5. On the other hand, we use $m_1 = 0.5$[kg] and $m_2 = 0.1$[kg] for each hole and peg with the inertia $J_1 = m_1 I_{3\times3}$[kgm^2] and $J_2 = \text{diag}(0.53, 0.53, 0.036)$[kgm^2]. For the consensus control of this virtual hole and peg, we use $(K^t_{1,2}, K^r_{1,2}) = (250, 250)$[N/cm] whereas we use $(K^t_i, K^r_i) = (300, 100)$ for the hand virtual proxies. Since the LAN with router has less than 5[ms] round-trip delay, as mentioned earlier, we impose artificial delay via the packet buffer. In the experiment, we use the artificial delay $T_d = \{25, 50, 100\}$[ms] to validate the stability of the passive simulation and the passive consensus control. For the damping (B^t, B^r), we use the larger one between the damping calculated from (5.3) and the critical damping (e.g., $B^t_i = \max(T_d K^t_i, 2\sqrt{m_i K^t_i})$, similar for the rotation part). Here, for the comparison with other open-source simulator (i.e., Bullet), we do not use the PSPM to eliminate further stabilizing effect during the energy modulation such as reharvesting which is hard to exactly calculate in the Bullet simulator.

We illustrate the snapshot of the remote user haptic interaction system with communication network in Fig. 5-3. The first column of the left figure is the virtual scene displayed for the user 1. On the other hand, the second column is the virtual scene displayed for the user 2 in an opposite view. During the experiment, the user 1 grab the virtual hole while the user 2 grab the virtual peg and inserting it to the hole. For every delay setting, the remote haptic interaction shows stable behavior. On the other hand, If non-passive simulation is used, it becomes unstable with increasing delay as shown in the right figure of Fig. 5-3. In the experiment, we use $(K^t_{1,2}, K^r_{1,2}) = (75, 75)$[N/cm] for the Bullet simulator, and increase the delay from 25[ms] to 100[ms] with damping injection calculated same way. For this, we impose some small force at initial time for small instance to the peg, and observe its behavior. As a result, the peg shows oscillatory behavior with small bound amplitude at $T_d = 50$[ms] and it becomes unstable with $T_d \geq 0$. Note that, one may expect this unstable behavior to be mitigated with large damping injection. However, the large damping can also destabilize the non-passive simulation. This clearly show that the enforcement of discrete-time passivity in PMI simulation and haptic rendering is crucial to ensure stable consensus between two users as in Fig. 5-3. This is because typical consensus algorithms, that can robustly ensure stability with communication delay, mandates open-loop system passivity, which is explicitly addressed by our proposed PMI-based haptic rendering.

Chapter 6

Conclusion and Future Works

Due to the broken physics, many behaviors which are not natural in real world occur in discrete-time simulation. For this reason, in this thesis, In this thesis, we develop a novel framework for haptic rendering and interactive simulation, that is based on the PMI (passive midpoint integrator), which is derived by directly enforcing passivity property in the discrete-time domain. In light of its passivity property, our PMI-based framework allows us: 1) to tune the virtual objects freely by exploring wide-range of parameters in a modular manner; 2) to stably simulate very light/stiff objects (e.g., for transparency) even with variable/slow integration steps (e.g., due to intermittent multi-point contact especially in real-time simulation); 3) to emulate such marginally-stable/lossless/energy-conserving behaviors as harmonic oscillation of flexible objects or elastic contact between rigid objects; and 4) to maintain stability of consensus control for multiuser haptic interaction under communication/information delay due to the open-loop passivity of local simulation. Further, our PMI-based framework provides implementation flexibility and convenience, by allowing for both the maximal and generalized coordinates, and also their combination, which are imperative to simulate some types of mechanical systems (e.g., multi-fingered UATD hand). We also devise both constraint-based and penalty-based contact formulation to fully incorporate the frictional multi-point contact with the Coulomb model.

In addition, we present the data-driven learning-based contact clustering framework, which consists of the multilayer perceptron (MLP) network and constraint-

based contact solver, to improve the accuracy (i.e., consistency with the real world experiment) of the contact simulation. The MLP network is designed to classify the active contact from the given contact point and/or normal set, and trained by the real experiment data with respect to the dynamics-related input (e.g., position, velocity, and external force etc.). This clustering is in fact crucial and employed in many real-time simulation, since it speeds up the contact simulation by reducing the computational cost and improves the numerical stability of the contact solver by avoiding possible over-constrained contact matrix.

The advantages of the PMI-based framework are experimentally demonstrated in Chap. 3.5, by using the three illustrative examples: haptic operation of peg-in-hole task; haptic interaction with flexible beam; and adaptive power-grasping by multi-fingered UATD hand. We also shows that the open-loop passivity of our PMI simulation plays a important role in stabilizing communication delay by employing it for the multiuser haptic interaction over UDP network in Chap. 5. On the other hand, performance evaluation of the data-driven learning-based contact clustering and comparison with the open-source robot simulator such as Bullet, ODE, and Vortex are proposed in Chap. 4.4, which shows that the proposed data-driven contact clustering significantly outperform Bullet, ODE, and the standard k-means clustering, whereas show meaningful accuracy improvement over Vortex.

Some future research directions for the proposed PMI-based haptic rendering and interactive simulation framework can be: 1) extension of the PMI-based framework to large-scale deformable objects, including cut-and-joining and fluidic simulation; 2) combining the constraint-stabilizing potential and PMI-LCP formulation while respecting non-iterativeness and passivity of the PMI (i.e., combining constraint-based and penalty-based multi-point contact formulation); and 3) development of open-source simulation platform based on our PMI-based simulation framework. On the other hand, for the data-driven learning-based contact clustering framework, future research will include: 1) further extending the proposed method for more complex contact scenario such as peg-in-hole assembly, and 2) generalizing it by incorporating collision detection information.

Bibliography

[1] M. Vidyasagar. *Analysis of nonlinear dynamic systems*. Prentice Hall, second edition, 1993.

[2] C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, pages 146–151, 1995.

[3] J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, volume 3, pages 140–145, 1995.

[4] D. J. Lee, M. Kim, and T. Qiu. Passive haptic rendering and control of lagrangian virtual proxy. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, pages 64–69, 2012.

[5] J. Bender, M. Müller, M. A. Otaduy, and M. Teschner. Position-based methods for the simulation of solid objects in computer graphics. In *Eurographics (STARs)*, pages 1–22, 2013.

[6] M. A. Otaduy, R. Tamstorf, D. Steinemann, and M. Gross. Implicit contact handling for deformable objects. *Computer Graphics Forum (Proceedings of Eurographics)*, 28(2):559–568, 2009.

[7] M. Ortega, S. Redon, and S. Coquillart. A six degree-of-freedom god-object method for haptic display of rigid bodies with surface properties. *IEEE Transactions on Visualization & Computer Graphics*, 13(3):458–469, 2007.

[8] J. Barbič and D. L. James. Six-dof haptic rendering of contact between geometrically complex reduced deformable models. *IEEE Transactions on Haptics*, 1(1):1–14, 2008.

[9] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Science & Business Media, 2006.

[10] M. B. Cline and D. K. Pai. Post-stabilization for rigid body simulation with contact and constraints. In *Proc. of IEEE Int'l Conf. on Robotics and Automation*, volume 3, pages 3744–3751, 2003.

[11] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. *ACM Transactions on Graphics*, 24(3):471–478, 2005.

[12] R. Weinstein, J. Teran, and R. Fedkiw. Dynamic simulation of articulated rigid bodies with contact and collision. *IEEE Transactions on Visualization & Computer Graphics*, 12(3):365–374, 2006.

[13] C. Duriez, F. Dubois, A. Kheddar, and C. Andriot. Realistic haptic rendering of interacting deformable objects in virtual environments. *IEEE Transactions on Visualization & Computer graphics*, 12(1):36–47, 2006.

[14] M. A. Otaduy and M. C. Lin. A modular haptic rendering algorithm for stable and transparent 6-dof manipulation. *IEEE Transactions on Robotics*, 22(4):751–762, 2006.

[15] H. Courtecuisse, Y. Adagolodjo, H. Delingette, and C. Duriez. Haptic rendering of hyperelastic models with friction. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, pages 591–596, 2015.

[16] CHAI-3D. Computer haptics and active interface, 2004.

[17] ODE. Open dynamics engine, 2001.

[18] VREP. Virtual robot experimentation platform, 2010.

[19] Bullet. Bullet real-time physics simulation, 2007.

[20] CM Labs. Vortex studio simulation platform, 2001.

[21] J. Chung and G. M. Hulbert. A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method. *Journal of Applied Mechanics*, 60:371–375, 1993.

[22] P. Krysl. Explicit momentum-conserving integrator for dynamics of rigid bodies approximating the midpoint lie algorithm. *International Journal for Numerical Methods in Engineering*, 63(15):2171–2193, 2005.

[23] P. Krysl. Direct time integration of rigid body motion with discrete-impulse midpoint approximation: explicit newmark algorithms. *Communications in Numerical Methods in Engineering*, 22(5):441–451, 2006.

[24] J. E. Colgate and G. Schenkel. Passivity of a class of sampled-data systems: application to haptic interfaces. *Journal of Robotic Systems*, 14(1):37–47, 1997.

[25] D. J. Lee and K. Huang. Passive-set-position-modulation framework for interactive robotic systems. *IEEE Transactions on Robotics*, 26(2):354–369, 2010.

[26] D. J. Lee and P. Y. Li. Passive bilateral control and tool dynamics rendering for nonlinear mechanical teleoperators. *IEEE Transactions on Robotics*, 21(5):936–951, 2005.

[27] D. J. Lee and M. W. Spong. Passive bilateral teleoperation with constant time delay. *IEEE Transactions on Robotics*, 22(2):269–281, 2006.

[28] S. P. Buerger and N. Hogan. Complementary stability and loop shaping for improved human-robot interaction. *IEEE Transactions on Robotics*, 23(2):232–244, 2007.

[29] J. M. Brown and J. E. Colgate. Minimum mass for haptic display simulations. In *ASME Dynamic Systems & Control Division*, pages 249–256, 1998.

[30] K. Huang and D. J. Lee. Consensus-based peer-to-peer control architecture for multiuser haptic interaction over the internet. *IEEE Transactions on Robotics*, 29(2):417–431, 2013.

[31] R. Rakhsha and D. Constantinescu. Passive shared virtual environment for distributed haptic cooperation. In *Proc. of Haptics Symposium*, pages 221–226, 2014.

[32] J. C. Simo and K. K. Wong. Unconditionally stable algorithms for rigid-body dynamics that exactly preserve energy and momemtum. *International Journal of Numerical Methods in Engineering*, 31(Jan.):19–52, 1991.

[33] J. C. Simo, N. Tarnow, and K. K. Wong. Exact energy-momentum conserving algorithms and symplectic schemes for nonlinear dynamics. *Computer Methods in Applied Mechanics & Engineering*, 100:63–116, 1992.

[34] D. Kuhl and M. A. Crisfield. Energy-conserving and decaying algorithms in non-linear structural dynamics. *International Journal of Numerical Methods in Engineering*, 45:569–599, 1999.

[35] K-J. Bathe. Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme. *Computers & Structures*, 85:437–445, 2007.

[36] J. M. Brown and J. E. Colgate. Passive implementation of multibody simulations for haptic display. In *Proc. of ASME Int'l Mechanical Engineering Congress & Exposition*, pages 85–92, 1997.

[37] D. J. Lee and K. Huang. On passive non-iterative varying-step numerical integration of mechanical systems for haptic rendering. In *Proc. of ASME Dynamic Systems & Control Conference*, 2008.

[38] J. E. Marsden and M. West. Discrete mechanics and variational integrators. *Acta Numerica*, 10:357–514, 2001.

[39] M. Kim, J. Kim, Y. Lee, and D. J. Lee. On the passivity of mechanical integrators in haptic rendering. In *Proc. of IEEE Int'l Conf. on Robotics & Automation*, pages 446–452, 2017.

[40] D. Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *Proc. of the Conference on Computer Graphics and Interactive Techniques*, pages 23–34, 1994.

[41] D. E. Stewart and J. C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *Int'l Journal for Numerical Methods in Engineering*, 39(15):2673–2691, 1996.

[42] J. Williams, Y. Lu, and J. C. Trinkle. A geometrically exact contact model for polytopes in multirigid-body simulation. *Journal of Computational and Nonlinear Dynamics*, 12(2):021001, 2017.

[43] N. Fazeli, E. Donlon, E. Drumwright, and A. Rodriguez. Empirical evaluation of common contact models for planar impact. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, pages 3418–3425, 2017.

[44] J. Hwangbo, J. Lee, and M. Hutter. Per-contact iteration method for solving contact dynamics. *IEEE Robotics and Automation Letters*, 3(2):895–902, 2018.

[45] E. Drumwright and D. A. Shell. Modeling contact friction and joint friction in dynamic robotic simulation using the principle of maximum dissipation. In *Algorithmic Foundations of Robotics IX*, pages 249–266. Springer, 2010.

[46] E. Todorov. A convex, smooth and invertible contact model for trajectory optimization. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, pages 1071–1076, 2011.

[47] E. Drumwright. A fast and stable penalty method for rigid body simulation. *IEEE Transactions on Visualization & Computer Graphics*, 14(1):240–231, 2008.

[48] M. Tang, R. Tong, Z. Wang, and D. Manocha. Fast and exact continuous collision detection with bernstein sign classification. *ACM Transactions on Graphics*, 33(6):186, 2014.

[49] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. In *Computer Graphics Forum*, volume 21, pages 279–287, 2002.

[50] K. Huang and D. J. Lee. Hybrid virtual-proxy based control framework for passive bilateral teleoperation over the internet. In *Proc. IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, pages 149–156, 2011.

[51] H. Delingette. Toward realistic soft-tissue modeling in medical simulation. *Proceedings of the IEEE*, 86(3):512–523, 1998.

[52] S. Hadap. Oriented strands - dynamics of stiff multi-body system. In *Proc. of ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pages 91–100, 2006.

[53] M. W. Spong, S. Hutchinson, and M. Vidyasaga. *Robot modeling and control*. John Wiley & Sons, Hoboken, NJ, 2006.

[54] J-J. E. Slotine and W. Li. On the adaptive control of robot manipulators. *International Journal of Robotics Research*, 6(3):49–59, 1987.

[55] D. J. Lee. Passive decomposition and control of nonholonomic mechanical systems. *IEEE Transactions on Robotics*, 26(6):978–992, 2010.

[56] D. J. Lee and K. Y. Lui. Passive configuration decomposition and passivity-based control of nonholonomic mechanical systems. *IEEE Transactions on Robotics*, 2(33):281–297, 2017.

[57] R. A Horn and C. R. Johnson. *Matrix analysis*. Cambridge university press, 2012.

[58] B. Leimkuhler and S. Reich. *Simulating hamiltonian dynamics*, volume 14. Cambridge university press, 2004.

[59] R. M. Murray, Z. Li, and S. S. Sastry. *A mathematical introduction to robotic manipulation*. CRC, Boca Ranton, FL, 1993.

[60] D. Q. Huynh. Metrics for 3d rotations: Comparison and analysis. *Journal of Mathematical Imaging and Vision*, 35(2):155–164, 2009.

[61] Steffen Knoop, Stefan Vacek, and Rüdiger Dillmann. Modeling joint constraints for an articulated 3d human body model with artificial correspondences in icp. In *IEEE-RAS Int'l Conf. on Humanoid Robots, 2005.*, pages 74–79, 2005.

[62] S Hasegawa and N Fujii. Real-time rigid body simulation based on volumetric penalty method. In *Proc. of IEEE Haptics Symposium*, pages 326–332, 2003.

[63] I. Peterlik, M. Nouicer, C. Duriez, S. Cotin, and A. Kheddar. Constraint-based haptic rendering of multirate compliant mechanisms. *IEEE Transactions on Haptics*, 4(3):175–187, 2011.

[64] M. Kolesnikov and M. Zefran. Energy-based 6-dof penetration depth computation for penalty-based haptic rendering algorithms. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, pages 2120–2125, 2007.

[65] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer Science & Business Media, 2008.

[66] J. E. Lloyd. Fast implementation of lemke's algorithm for rigid body contact simulation. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, pages 4538–4543, 2005.

[67] R. W. Cottle, J.-S. Pang, and R. E. Stone. *The linear complementarity problem*, volume 60. Siam, 2009.

[68] R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics & Automation*, 15(3):465–474, 1999.

[69] P. Mitra and G. Niemeyer. Dynamic proxy objects in haptic simulations. In *Proc. of IEEE Conf. on Robotics, Automation & Mechatronics*, pages 1054–1059, 2004.

[70] G. Yu, D. Wang, and Y. Zhang. Accelerating optimization-based haptic rendering by parallel quadratic programming method. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, pages 4499–4504, 2013.

[71] G. Cirio, M. Marchal, M. A. Otaduy, and A. Lécuyer. Six-dof haptic interaction with fluids, solids, and their transitions. In *Proc. of World Haptics Conference*, pages 157–162. IEEE, 2013.

[72] D. J. Lee. Extension of colgate's passivity condition to variable-rate haptics. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, pages 1761–1766, 2009.

[73] D. J. Lee and P. Y. Li. Passive decomposition of mechanical systems with coordination requirement. *IEEE Transactions on Automatic Control*, 58(1):230–235, 2013.

[74] W. H. Chen, D. J. Ballance, P. J. Gawthrop, and J. O'Reilly. A nonlinear disturbance observer for robotic manipulators. *IEEE Transactions on Indurstrial Electronics*, 47(4):932–938, 2000.

[75] A. Gupta and M. K. O'Malley. Disturbance-observer-based force estimation for haptic feedback. *Journal of Dynamic Systems, Measurement, and Control*, 133(1):014505, 2011.

[76] J. J. E. Slotine and W. Li. *Applied nonlinear control*, volume 199. Prentice-Hall Englewood Cliffs, NJ, 1991.

[77] R. W. Daniel and P. R. McAree. Fundamental limits of performance for force reflecting teleoperation. *International Journal of Robotics Research*, 8:811–830, 1998.

[78] T. Iwasaki, S. Hara, and H. Yamauchi. Dynamical system design from a control perspective: finite frequency positive-realness approach. *IEEE Transactions on Automatic Control*, 48(8):1337–1354, 2003.

[79] M. Franken, S. Stramigioli, S. Misra, C. Secchi, and A. Macchelli. Bilateral telemanipulation with time delays: a two-layer approach combining passivity and transparency. *IEEE Transactions on Robotics*, 27(4):741–756, 2011.

[80] Y. Lee, M. Kim, Y. Lee, J. Kwon, Y.-L. Park, and D. J. Lee. Wearable finger tracking and cutaneous haptic interface with soft sensors for multi-fingered virtual manipulation. *IEEE/ASME Transactions on Mechatronics*, 2018.

[81] M. Kim and D. J. Lee. Improving transparency of virtual coupling for haptic interaction with human force observer. *Robotica*, 35(2):354–369, 2017.

[82] D. Prattichizzo, F. Chinello, C. Pacchierotti, and M. Malvezzi. Towards wearability in fingertip haptics: a 3-dof wearable device for cutaneous force feedback. *IEEE Transactions on Haptics*, 6(4):506–513, 2013.

[83] S. S. Rao and F. F. Yap. *Mechanical vibrations*, volume 4. Addison-Wesley New York, 1995.

[84] J. E. Colgate and J. M. Brown. Factors affecting the z-width of a haptic display. In *Proc. IEEE Int'l Conf. on Robotics & Automation*, pages 3205–3210, 1994.

[85] L. U Odhner, L. P. Jentoft, M. R. Claffee, N. Corson, Y. Tenzer, R. R. Ma, M. Buehler, R. Kohout, R. D. Howe, and A. M. Dollar. A compliant, underactuated hand for robust manipulation. *International Journal of Robotics Research*, 33(5):736–752, 2014.

[86] R. Ozawa, H. Kobayashi, and K. Hashirii. Analysis, classification, and design of tendon-driven mechanisms. *IEEE Transactions on Robotics*, 30(2):396–410, 2014.

[87] D. Baraff. Linear-time dynamics using lagrange multipliers. In *Proc. of the Conference on Computer Graphics and Interactive Techniques*, pages 137–146. ACM, 1996.

[88] A. Rocchi, B. Ames, Z. Li, and K. Hauser. Stable simulation of underactuated compliant hands. In *Proc. of the IEEE Int'l Conf. on Robotics and Automation*, pages 4938–4944, 2016.

[89] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651—-666, 2010.

[90] Gazebo. Gazebo simulator, 2017.

[91] S. S. Haykin. *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River, NJ, USA:, 2009.

[92] E. Rohmer, S. P. Singh, and M. Freese. V-rep: A versatile and scalable robot simulation framework. In *Proc. of IEEE/RSJ Int'l Conf. on Intelligent Robots & Systems*, pages 1321–1326, 2013.

[93] Y. R. Stürz, L. M Affolter, and R. S. Smith. Parameter identification of the kuka lbr iiwa robot including constraints on physical feasibility. *IFAC-PapersOnLine*, 50(1):6863–6868, 2017.

[94] C. Schuwerk, X. Xu, R. Chaudhari, and E. Steinbach. Compensating the effect of communication delay in client-server–based shared haptic virtual environments. *ACM Transactions on Applied Perception (TAP)*, 13(1):5, 2015.

[95] Clemens Schuwerk, Xiao Xu, and Eckehard Steinbach. On the transparency of client/server-based haptic interaction with deformable objects. *IEEE transactions on haptics*, 10(2):240–253, 2017.

[96] R. Rakhsha, D. Constantinescu, and Y. Shi. Passive shared virtual environment for haptic cooperation. *Autonomous Robots*, pages 1–16, 2018.

[97] R. Olfati-Saber and R. M. Murray. Consensus problems in networks of agents with switching topology and time-delays. *IEEE Transactions on automatic control*, 49(9):1520–1533, 2004.

초 록

수동성기반 다중접촉 햅틱렌더링 및 인터렉티브 시뮬레이션 프레임워크 개발

김 명 신
기계항공공학부 기계전공
The Graduate School
Seoul National University

본 논문에서는 수동성 기반의 다중접촉 햅틱렌더링 및 실시간 시뮬레이션 기법을 제안한다. 제안된 기법은 1) 일반화좌표계의 비선형 라그랑지 동역학의 이산시간 수동성을 보장하는 Passive Midpoint Integration, 2) 최대좌표계의 SE(3) 강체들과 그들 사이의 구속력을 수동성을 근사적으로 보장하도록 하는 Passive Midpoint Integration, 3) 물체들 사이의 다중접촉을 풀기 위한 컨스트레인트 기반 및 페널티기반의 다중접촉 시뮬레이션 기법, 4) 가상커플링을 이용한 햅틱렌더링기법, 및 마지막으로 5) 다중접촉의 정확도 향상을 위한 실험데이터 및 기계학습기반의 접촉 클러스터링 기법으로 구성된다.

제안된 기법의 유효성을 검증하기 위하여 진동빔 시뮬레이션, 부족구동 텐던로봇손, 다중접촉 펙인홀 작업 및 통신시간지연하의 원격사용자간의 협동 펙인홀 햅틱 상호작용에 대한 실험 결과를 제시하며, 다중접촉 시뮬레이션의 성능비교평가 결과를 제시한다.

주요어: 수동성, 햅틱렌더링, 실시간 시뮬레이션, 다중접촉
학 번: 2012-20651