Haptic manipulation of virtual linkages with closed loops

Haptic Manipulation of Virtual Linkages with Closed Loops

M.A. Beenakkers

DCT 2006-123

Traineeship report

Coaches: D. Constantinescu

Supervisor: M.Steinbuch

Technische Universiteit Eindhoven
Department of Mechanical Engineering
Control Systems Technology Group

Eindhoven, October, 2006

Contents

Abstract

Nowadays, surgeons located in Amsterdam can operate patients in Vancouver using visual and touch/force feedback received through a network connection. A new challenge currently facing engineers is to enable surgeon students to practice surgeries on computer simulated organs rather than on cadavers or patients. These examples are applications of haptic manipulations of remote and virtual objects, i.e., manipulations that involve the user's sense of touch.

The Assistive Robotics and Mechatronics Systems (ARMS) Laboratory in the Department of Mechanical Engineering at the University of Victoria carries out research in haptic rendering and control of user interaction within virtual environments (VEs). The goal is to provide users with realistic haptic (force) feedback while they manipulate arbitrary virtual objects. Presently, the ARMS Laboratory is interested in expanding the set of virtual objects that the users can manipulate from single rigid objects and serial virtual mechanisms (VMs) to include VMs with closed loops. Haptic manipulation of VM with closed loops is beneficial in virtual prototyping applications and in virtual reality-based training for minimally invasive procedures. Increasingly more often, minimally invasive surgeries are performed using robots like the Da Vinci system. Therefore, the haptic training system must enable users to manipulate sophisticated tools, including those caried by linkages with closed loops.

This report addresses the current focus of the ARMS Laboratory. The reported work proposes a haptic rendering technique that allows users to manipulate VMs with closed loops. Realistic manipulation of such VMs requires users to feel the loop closure constraints. These constraints affect the inertia of the VM and the user's freedom of motion. The report proposes a technique for computing the inertia of VM with closed loops and the directions of motion resisted by the virtual joints at the user's hand. The new technique readily integrates with the control architecture of an existing haptic setup and can be used to enable users to operate VMs with closed loops from any user-selected link.

The report describes the implementation of a closed loop mechanism in the existing haptic interaction system. The introduction chapter offers further insight into the field of haptics. The dynamics of mechanisms with closed loops are derived in Chapter 2. The implementation of VMs in the haptic setup is detailed in Chapter 3, focusing on the requirements imposed on the simulation by the loop closure constraints. The control structure used for haptic rendering is overviewed in Chapter 4, followed by the method proposed for computing the VM inertia at the user's hand and the directions of motion restricted by the virtual joints in Chapter 5. Experimental results are presented in Chapter 6 to validate the methods developed in this work and their implementation in the haptic setup. The report ends with conclusions and recommendations for future work.

The work presented in this report has been submitted to the 2007 IEEE International Conference on Robotics and Automation. The submission is appended to the report.

Chapter 1

Introduction

1.1 What is haptics

Psychophysics introduced the word haptics in the early 20th century. In psychophysics, haptics describes the subfield of study which addresses human touch-based perception and manipulation. In the 70's and 80's, robotics research also began to focus on perception and manipulation by touch. The development of a human-like robotic hand was investigated. The robotic hand was too complex to build at the time. Two communities were created in the 80's: one group sought to understand the human hand; another group tried to build a robotic hand inspired by human abilities. In the 90's, haptics was endowed with yet another meaning. Computers became widely available and their computational power allowed virtual environments (VEs) to be simulated and displayed at interactive rates (for example: computer games). Computer haptics emerged as the filed of research that enables users to interact with virtual environments through touch. As described above, haptics can be divided in three areas [25]:

1. Human haptics, which is the study of human sensing and manipulation through touch;

2. Machine haptics, which encompasses the design, construction, and use of machines to replace or augment the human touch;

3. Computer haptics, which includes the algorithms and software associated with generating and rendering the touch and feel of virtual objects (analogous to computer graphics).

Researchers in fields ranging from robotics to neuroscience are presently advancing the frontiers of haptics.

The topic of this report is haptic rendering which is a subfield of computer graphics. Haptics refers to touch interactions, and rendering refers to the information sent to the user. The information has to be sufficient to give the user realistic touch perceptions in VEs.

VE applications strive to simulate objects with which users can interact. In real life 'interaction' means that users feel, see, taste, smell and hear the environment. In the VE, the number of senses involved in the interaction is rather small. The senses most commonly employed in VEs are vision, hearing and more recently touch. The basic structure of a VE application incorporating visual, auditory and haptic feedback is shown in Figure 1.1.

In VE applications the simulation engine computes the VE behavior in time. The visual, auditory and haptic rendering algorithms compute the graphics, sound and force feedback to the user. The transducers convert the discrete computer signals to the visual, audio and force signals received by users. Audio and visual feedback is typically provided to users through speakers and displays. These are unidirectional computer interfaces which only send signals to users. In contrast, the computer interface which adds touch to VEs, called the haptic device, is bidirectional. It sends and receives signals to and from users. Depending on their purpose, different haptic setups require different haptic interfaces. Figure 1.2 shows various haptic devices of increasing complexity.

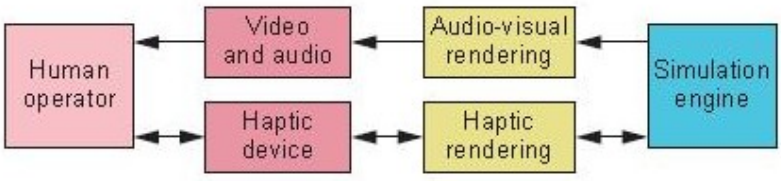


Figure 1.1: Basic architecture for a virtual reality application incorporating visual, auditory, and haptic feedback. [Reproduced with permission from [24]]

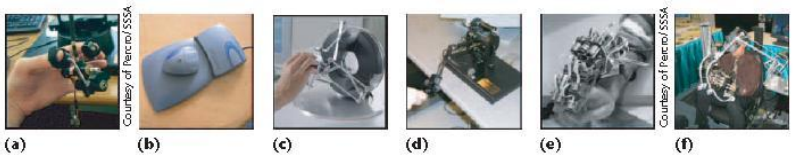


Figure 1.2: Examples of increasingly more complex haptic devices: (a) force-reflecting gripper, (b) Logitech's Wingman force feedback mouse, (c) Force Dimension's Omega haptic device, (d) SensAble's Phantom haptic device, (e) the Hand Force Feedback exoskeleton, and (f) Immersion's Haptic Workstation. [Reproduced with permission from [24]]

1.2 Experimental setup

The purpose of the experimental setup used in this report is to allow arbitrary body interaction within planar VEs. The haptic interface [25] comprises two five-bar mechanisms mounted in parallel. This design allows the device handle to translate in the horizontal plane and to rotate about a vertical axis. The three degrees of freedom (DOF) of the haptic interface (two translations and one rotation) are controlled by four electric motors mounted on the base joints. Position encoders are attached to motor shafts. The position of the user's hand, i.e., the position of the device handle, is known from the encoder readings and the kinematics of the haptic interface. Vortex^{tm}, a physics-based simulation engine from CM Labs Inc., is used to simulate and display the VE. The experimental setup uses two computers: (i) one computer runs Windows 2000 and simulates the VE at visually interactive rates of 10 to 60 Hz; and (ii) a second computer runs VxWorks and computes the haptic control and rendering algorithms at 512Hz. This setup is shown in Figure 1.3.

1.3 Challenges in haptics

This section briefly overviews the following key challenges in computer haptics: (i) discrete equivalents of continuous-time systems; (ii) virtual walls; (iii) system dynamics; and (iv), different sample times in the simulation and haptics engines. These challenges are currently topics of active research and a discussion of existing solutions is beyond the scope of the present report.

1.3.1 Discrete equivalents

In haptics, the discrete implementation of a continuous-time system is needed to simulate a physical system. Discretization is imposed by the computer implementation of the control and simulation algorithms and sometimes by the use of discrete amplifiers. The discrete equivalent of a transfer function via numerical integration is obtained by applying one of the following integration techniques to the differential

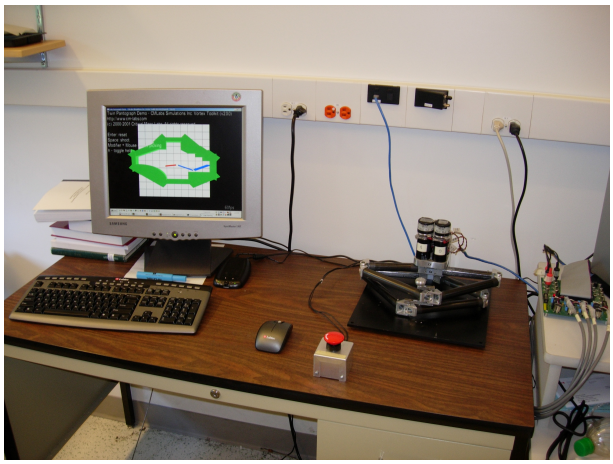


Figure 1.3: The haptic setup used in the present report

equation of the system:

$$\text{Forward rectangular rule:} \quad \dot{x}(k) = \frac{x(k+1) - x(k)}{T} \tag{1.1}$$

$$\text{Backward rectangular rule:} \quad \dot{x}(k+1) = \frac{x(k+1) - x(k)}{T} \tag{1.2}$$

$$\text{Trapezoid rule:} \quad \frac{\dot{x}(k+1) - \dot{x}(k)}{2} = \frac{x(k+1) - x(k)}{T} \tag{1.3}$$

In Equations (1.1), (1.2) and (1.3), x is the state of the system at time k, and T is the sample time. The mapping of the stable region in the s-plane (i.e., the left half plane) onto the z-plane is illustrated in Figure 1.4 for each rule. Note that stable continuous-time systems may become unstable when discretized via the forward rectangular rule, as done in haptics.

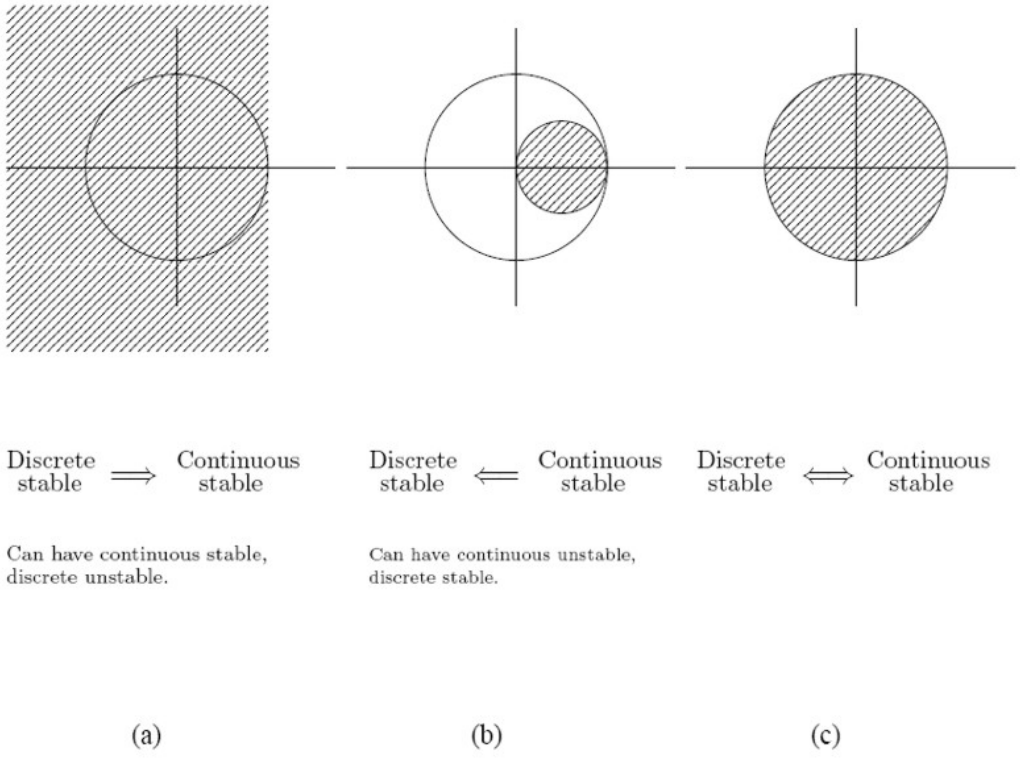


Figure 1.4: The image of the stable region in s-plane under three transformations: (a) forward rectangular; (b) backward rectangular; and (c) trapezoidal. [Reproduced with permission from [13]]

1.3.2 Virtual walls

Virtual walls are commonly used in haptics. Walls define for example the working space of a free moving body. Physical walls are passive, i.e., they generate no energy when bodies contact them. Virtual walls in haptics are active, because they are typically implemented as spring-damper systems in discrete time. A virtual spring can be used to gain insight. An ideal physical spring is a lossless system. It returns the same energy during extension as put into the spring during compression. However, a virtual spring is implemented in discrete time. The force provided by the virtual spring does not increase smoothly with deflection. Instead, the force is "held" at a constant value during each simulation step. The zero order hold generates energy, because the average force (and work) during squeezing is less than the average force during releasing. The virtual spring always generates energy. See Figure 1.5 for an intuitive graphical representation.

In Figure 1.5, the shaded area is a measure of the energy generated by the virtual wall. The stiffness of the virtual spring (K) represents the stiffness of the wall. Another wall characteristic is damping (B), which must be sufficient to prevent noticeable oscillations. Unfortunately, increasing B causes high frequency oscillations that users often report as a feeling of "rumble". This is because the discrete implementation of virtual damping can produce energy, usually under high frequency excitation [3].

1.3.3 Dynamic systems

The models of virtual objects that currently can be implemented in haptics are quite simple compared to objects in the real world. The development of computationally efficient models and interaction techniques is still a challenge, even with the current rate of increase in processing speeds. The research in dynamic models and interaction techniques is important because faster and more accurate haptic simulations are

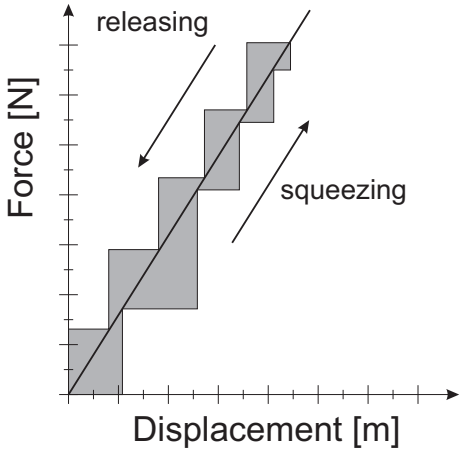


Figure 1.5: Representation of a discretized virtual spring. (Reproduced with permission from [3])

key for providing more realistic touch sensations to users.

The multibody systems presently implemented in haptic setups are simple. They are typically open loop, and are computed in independent coordinates, such that no constraint equations (and constraint stabilization technique) are needed, as discussed in further detail in Chapter 2. The absence of constraint equations and the system simplicity save computation time.

1.3.4 Different sample times

The experimental setup used in this report employs a haptics processor distinct from the simulation processor. The separation relieves the hard real time requirements imposed by the haptic controller on the VE simulation. A drawback is that the two engines run at different sampling times. The VE simulation runs at variable sample times of 10 to 50 Hz, and the haptic engine runs at a fixed sample time of 512Hz. The setup needs an additional synchronization mechanism between the two engines.

1.4 Objective

The objective of the present work is to expand the set of VEs that the users can touch and manipulate using the experimental setup shown in Section 1.2. Specifically, the goal is to enable users to operate VMs with closed loops in addition to single objects and serial virtual linkages.

Realistic manipulation of VMs with closed loops requires users to feel the influence of the loop closure constraints on their the VM inertia and on the users' freedom of motion. Therefore, this report proposes a method to compute: (i) the VM inertia reflected at the user's hand; and (ii) the directions of motion restricted by the joints of the VM. These inertia and directions of motion are incorporated in the control architecture of the experimental setup. The ability to render realistic manipulations of VM with closed loops using the derived inertia and directions of restricted motion is validated via haptic manipulation of a planar closed loop VM.

Chapter 2

Dynamics of closed-loop multibody systems

The VE used to illustrate the haptic rendering techniques proposed in this report consists of a closed-loop virtual mechanism (VM) moving freely inside a rigid enclosure, as depicted in Figure 2.1. Therefore, the dynamics of multibody systems with closed-loops are overviewed in this chapter. The dynamics of the closed-loop VM implemented in the experimental setup are discussed in Chapter 3.

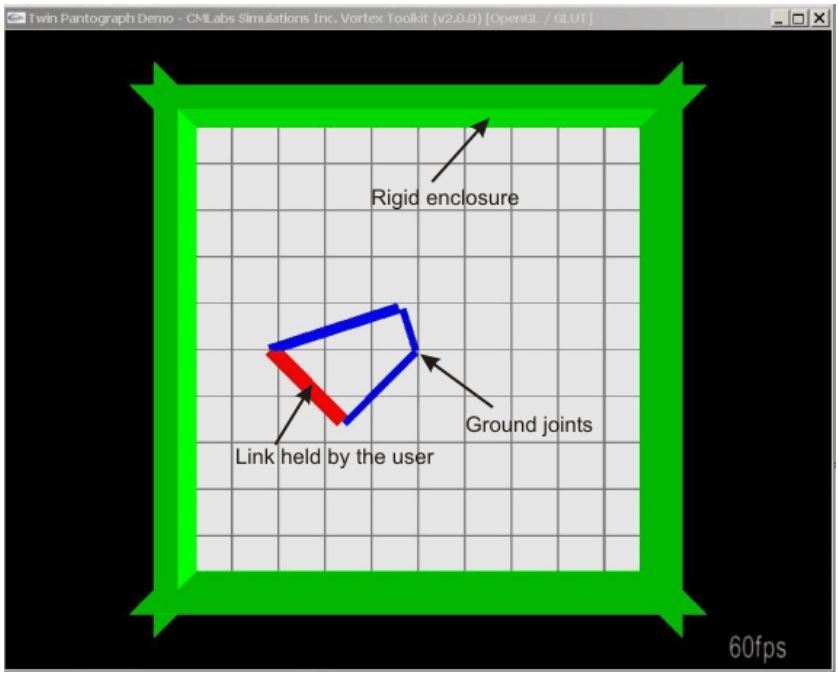


Figure 2.1: The closed-loop virtual mechanism (VM) implemented in the experimental test bed.

2.1 Multibody system kinematics

This section presents the dynamics of closed loop multibody systems. The system considered in this report has k rigid bodies. Free rigid bodies can have spatial or planar motion. A free rigid body in spatial motion has six DOFs, three translations and three rotations. A free rigid body in planar motion has three DOFs, two translations and one rotation. The haptic setup used in this report comprises a planar haptic interface and can render only planar VEs. Therefore, the multibody system considered in this report has planar motion. The translations are along the x- and y-axes, and the rotation is about the z-axis. A rigid body can have m kinematic constraints. Kinematic constraints express relations between

a body and its environment. Constrained rigid bodies have less DOFs than free rigid bodies. Constraints reduce the DOFs of rigid bodies.

Dynamic systems can be described in absolute or in relative coordinates. Absolute coordinates describe the position of each body with respect to a common reference frame. Relative coordinates describe the position of each body with respect to an adjacent frame, that is, with respect to a frame attached to an adjacent body. These types of coordinates are illustrated in Figure 2.2.

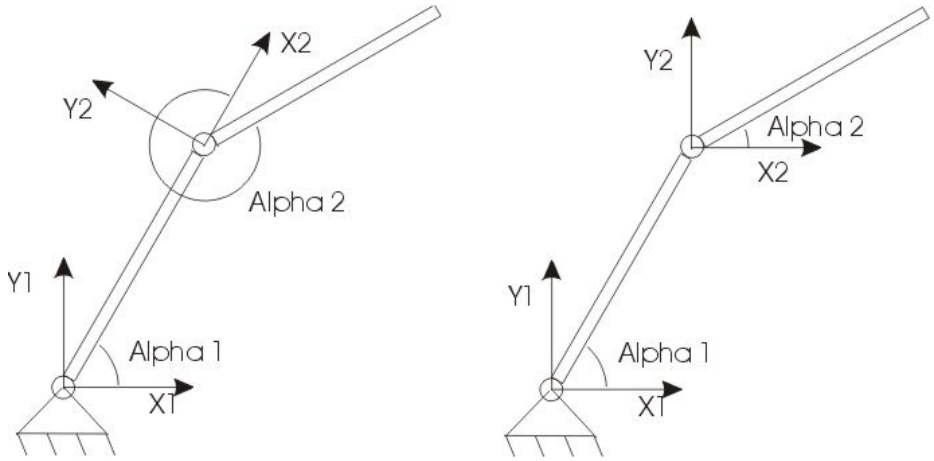


Figure 2.2: Coordinate systems used to describe dynamic systems: relative coordinates are used in the left hand side figure; absolute coordinates are used in the right hand side figure.

A constrained dynamic system can be described in less coordinates than absolute or relative coordinates. Generically, the coordinates used to describe the system are called generalized coordinates. The generalized coordinates can be dependent or independent. The system kinematics describes the motion of the system without considering the forces and torques that act on the system and cause its motion.

2.2 Multibody system dynamics

Two commonly used techniques for deriving the dynamics of multibody systems are the Newton-Euler and the Lagrange method. This report uses the Lagrange method, in which the equations of motion of the system are:

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \underline{\dot{q}}}\right) - \frac{\partial L}{\partial \underline{q}} + \underline{W}^T \underline{\lambda} = \underline{Q}^{nc} \tag{2.1}$$

In Equation (2.1), $L = T - V$ is the system Lagrangian, and is equal to the difference between the kinetic energy T and the potential energy V of the system; t is time; $\underline{W}$ is the Jacobian of the constraints; $\underline{\lambda}$ is the vector of Lagrange multipliers (magnitudes of the unknown constraint forces); $\underline{Q}^{nc}$ are the external forces and torques applied on the system; $\underline{q} = (q_1 \dots q_n)^T$ and $\underline{\dot{q}} = (\dot{q}_1 \dots \dot{q}_n)^T$ are the generalized position and velocity vectors; and the superscript ^{T} denotes matrix transposition. All Jacobians used in this report are derivatives of vectors with respect to the generalized coordinates. The applied forces $\underline{Q}$ and the Jacobian of the constraints $\underline{W}$ will be discussed in further detail in Sections 2.2.2 and 2.2.3, respectively. The number of scalar equations in Equation (2.1) is equal to the number of coordinates used to describe the system, n. After substituting $L = T - V$, Equation (2.1) becomes:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \underline{\dot{q}}}\right) - \frac{\partial T}{\partial \underline{q}} + \frac{\partial V}{\partial \underline{q}} = -\underline{W}^T \underline{\lambda} + \underline{Q}^{nc} \tag{2.2}$$

Equations (2.1) or (2.2) together with the m algebraic constraint equations $\underline{h}(\underline{q}) = \underline{0}$ form a system of $m + n$ differential algebraic equations (DAEs). This system has n unknown accelerations and m unknown Lagrange multipliers (constraint or reaction forces). Integration of DAEs is unsuited for real-time performance. Therefore, the DAEs are transformed into ordinary differential equations (ODEs) via differentiating the constraint equations twice with respect to time $\ddot{\underline{h}}(\underline{q}) = \underline{0}$. In addition, expressing Equations (2.1) or (2.2) in generalized coordinates leads to:

$$\underline{D}(\underline{q})\ddot{\underline{q}} + \underline{C}(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} + \underline{g}(\underline{q}) = \underline{J}_h^T \underline{F}_h + \underline{W}^T \underline{\lambda} \tag{2.3}$$

In Equation (2.3): $\underline{D}(\underline{q})$ is the mass (or inertia) matrix of the multibody system; $\underline{C}(\underline{q}, \dot{\underline{q}})\dot{\underline{q}}$ groups the centripetal and Coriolis terms; $\underline{g}(\underline{q})$ encompasses joint flexibility and gravitational effects; $\underline{F}_h$ are time dependent non-conservative forces and torques (due to damping and the user); $\underline{J}_h$ is the Jacobian of the position of the user-applied force (i.e., the hand Jacobian); $\ddot{\underline{q}} = (\ddot{q}_0 \dots \ddot{q}_n)$ is the generalized acceleration vector. In this report, gravitational and flexibility effects are neglected because the system under consideration moves in the horizontal plane, without flexibility. The $\underline{D}$ and $\underline{C}$ matrices are derived in Section 2.2.1, and $\underline{J}_h$ is discussed in Section 2.2.3.

2.2.1 Mass-matrix and C-matrix

Equation (2.3) can be derived from Equation (2.2) in several steps. In a first step, the mass matrix is derived via:

$$\underline{D}(\underline{q})\ddot{\underline{q}} = \frac{d}{dt}\left(\frac{dT}{d\dot{\underline{q}}}\right) \tag{2.4}$$

The C-matrix can be computed according to:

$$\underline{C}(\underline{q}, \dot{\underline{q}})\dot{\underline{q}} = -\left(\frac{dT}{d\underline{q}}\right) \tag{2.5}$$

Equation (2.4) can be rather complex to calculate when the kinetic energy is expressed in absolute coordinates. An alternative way to compute $\underline{D}$ exists which is commonly used in robotics. In the robotics method, the system is described in generalized coordinates and the velocity of the centre of mass of body i, $\underline{\dot{r}}_{cm,i}$, as a function of the generalized velocities is:

$$\underline{\dot{r}}_{\,cm,i} = \underline{J}_{\,cm,i}\ \underline{\dot{q}} \tag{2.6}$$

In Equation (2.6) $\underline{\dot{r}}_{\,cm,i} = [\dot{x}\ \dot{y}\ \dot{\theta}]^T$ is the velocity of body i in Cartesian coordinates, $\underline{J}_{\,cm,i}$ is the Jacobian of the position of the centre of mass of body i. After substituting Equation (2.6) in the kinetic energy of body i, the kinetic energy of body i becomes:

$$\begin{aligned} T_i \;=\; & 1/2\ m_i \dot{\vec{r}}_{\,cm,i} \cdot \dot{\vec{r}}_{\,cm,i} + 1/2\ \vec{\omega}_{\,i} \cdot (I_{\,cm,i} \cdot \vec{\omega}_i) = \\ & 1/2\ \underline{\dot{q}}^T \cdot (m_i \underline{J}^T_{cm,i} \underline{J}_{\,cm,i}) \cdot \underline{\dot{q}} + 1/2\ \underline{\omega}_i^T \underline{I}_{\,cm.i} \underline{\omega}_i = \\ & 1/2\ \underline{\dot{q}}_i^T \cdot \underline{\Lambda}_i \cdot \underline{\dot{q}}_i + 1/2\ \underline{\omega}_i^T I_{\,cm,i} \underline{\omega}_i \quad i = 1, \dots, k \end{aligned} \tag{2.7}$$

In Equation (2.7), $\underline{I}_{\,cm,i}$ is the inertia matrix of body i with respect to its centre of mass. The matrices $\underline{\Lambda} = m \cdot \underline{J}^T \cdot \underline{J}$ and $\underline{I}_{cm}$ are symmetric. The kinetic energy of the system is the sum of kinetic energies of all bodies in the system:

$$T = \sum_{i=1}^{k} T_i \tag{2.8}$$

Hence, the D-matrix can be computed as:

$$\underline{D}(\underline{q})\ddot{\underline{q}} = \frac{d}{dt}\left(\frac{d\underline{T}}{d\dot{\underline{q}}}\right) = \frac{\underline{\Lambda}\dot{\underline{q}}}{dt} + \frac{d\left(1/2\ \underline{\omega I}_{\ cm}\underline{\omega}\right)}{dt} = (\underline{\Lambda} + \underline{I})\ddot{\underline{q}}, \tag{2.9}$$

or:

$$\underline{D}(\underline{q}) = \underline{\Lambda}(\underline{q}) + \underline{I}, \tag{2.10}$$

and the C-matrix becomes:

$$\underline{C}_{kj} = \sum_{i=1}^{\#bodies} c_{ijk}(\underline{q})\dot{\underline{q}}, \tag{2.11}$$

where:

$$c_{ijk} = 1/2\ \left(\frac{\partial \underline{D}_{\ kj}}{\partial \underline{q}_i} + \frac{\partial \underline{D}_{\ ki}}{\partial \underline{q}_j} + \frac{\partial \underline{D}_{\ ij}}{\partial \underline{q}_k}\right) \quad i,j,k = 1,\ldots,n. \tag{2.12}$$

2.2.2 Kinematic constraints

The constraints of the system depend on the system configuration, and are caused by joints, non-slip conditions, etc. If the constrains can be expressed as functions of configuration and time only:

$$\underline{\mathbf{h}}(\mathbf{q},\mathbf{t}) = \begin{pmatrix} \underline{h}_1(\underline{q},t) \\ \vdots \\ \underline{h}_m(\underline{q},t) \end{pmatrix}, \tag{2.13}$$

they are called holonomic. If the constraints are also time independent, they are called scleronomic. Only scleronomic constraints are considered in this report. The constraint equations must hold for any admissible virtual displacement (in other words, the constraint forces do not work along the admissible displacements):

$$\delta\underline{h} = \underline{W}^T\delta\underline{q} = \underline{0} \tag{2.14}$$

In Equation (2.14), $\underline{W} = \frac{d\underline{h}}{d\underline{q}}$ is the Jacobian of the constraints that was introduced in Equation (2.1). Further details can be found in [26].

2.2.3 Non-conservative forces

The externally applied forces and torques and the friction acting on the system are non-conservative generalized forces ($\underline{Q}_{nc}$). Friction depends on velocity. Non-conservative forces may work. The velocities of the points of application of the system forces have to be compatible with the constraints. Therefore the absolute position and orientation of the body at the point where a force is applied needs to be computed. The non-conservative forces can be found by:

$$\underline{Q}^{nc} = \sum_{i=1}^{n_F} \frac{d\underline{r}_i^{nc}}{d\underline{q}} \cdot \underline{F}_i^{nc}. \tag{2.15}$$

In Equation (2.18):

$$\underline{\mathbf{r}}_{\mathbf{i}}^{\mathbf{nc}} = \begin{pmatrix} x_h \\ y_h \\ \theta_h \end{pmatrix} \tag{2.16}$$

and:

$$\underline{\mathbf{F}}_{\mathbf{i}}^{\mathbf{nc}} = \begin{pmatrix} F_x \\ F_y \\ T_z \end{pmatrix} \tag{2.17}$$

and n_F is the number of applied non-conservative forces. Note that the forces considered in this report have only three components, because the closed-loop VM manipulated by the user move in the horizontal plane. Moreover, only one external force is applied to the VM ($n_F = 1$). This external force is the user-applied (also called the hand) force F_h. The hand position and orientation are given by r_h. The hand Jacobian J_h introduced in Equation (2.3) is derived via:

$$\underline{Q}^{nc} = \frac{d\underline{r}^{nc}}{d\underline{q}} \cdot F^{nc} = \frac{d\underline{r}_h}{d\underline{q}} \cdot F_h = \underline{J}_h \cdot \underline{F}_h \qquad \underline{J}_h = \frac{d\underline{r}_h}{d\underline{q}} \tag{2.18}$$

Chapter 3

Virtual Mechanism Implementation

The VM implemented in the experimental haptic setup to validate the haptic rendering techniques proposed in this report is a closed loop VM as shown in Figure 3.1. The links of the VM have length L_i, mass m_i and inertia J_i, all given in the international system of units. The fraction $f_{cm,i}$ defines the distance to the COM of the link with respect to the adjacent joint as a percentage of the entire link length, $l_{cm,i} = f_{cm,i} \cdot l_i$, with $i = 1, \ldots, k$. Q_j defines the j^{th} linkage in the system, with $j = 1, \ldots, p$. The system dynamics are derived in relative dependent coordinates, and the joint angles are measured with respect to the adjacent body.

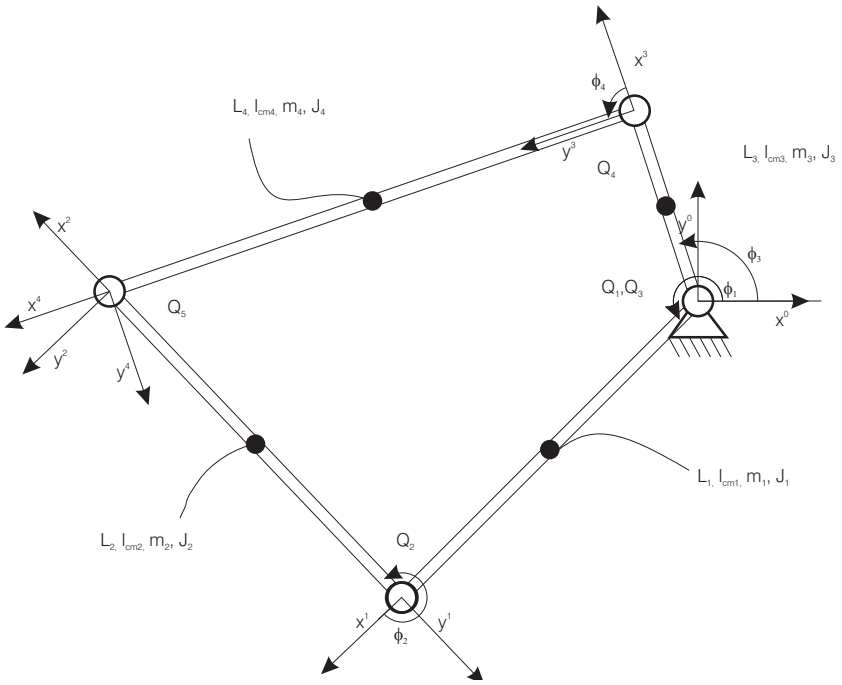


Figure 3.1: Closed loop VM used to experimentally validate the haptic rendering techniques proposed in this report.

The dynamics of the VM are derived as in Equation (2.3). Because the closed loop dynamics are formulated in dependent coordinates, additional constraint equations are needed to solve the Lagrangian dynamics given in Equation (2.3). For holonomic constraints, the additional constraint equations are:

$$\underline{W}^T(\underline{q}, t)\ddot{\underline{q}} + \underline{\bar{w}}(\underline{q}, \dot{\underline{q}}, t) = \underline{0}, \tag{3.1}$$

where:

$$\underline{\bar{w}}(\underline{q},\dot{\underline{q}},t) = \left(\frac{\partial \underline{W}^T(\underline{q},t)}{\partial t} + \frac{\partial \underline{W}^T(\underline{q},t)\dot{\underline{q}}}{\partial \underline{q}}\right)\cdot \dot{\underline{q}}. \tag{3.2}$$

The matrices used in Equations (2.3), (3.1) and (3.2) in the current implementation are given in Appendix A.

The dynamics of the closed-loop VM are simulated in Simulink® using the model shown in Figure 3.2. This model includes the equations needed to compute $\ddot{\underline{q}}$ and $\underline{\lambda}$, the structure of the haptic and graphic blocks is shown in Figure 3.5 and the controller is a PD-controller .

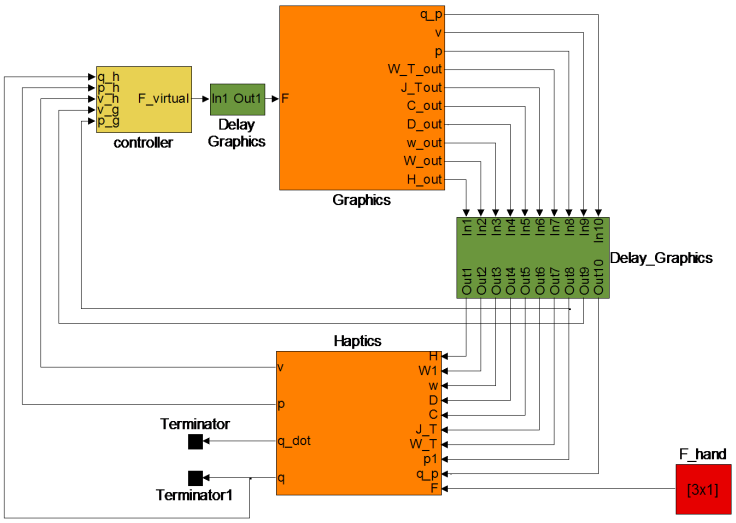


Figure 3.2: Simulink® model implemented in Matlab® for simulating user's haptic interaction with the VM depicted in Figure 3.1.

Figure 3.3 illustrates that the loop closure constraint errors diverge from zero as the simulation progresses. This is because the differential equation $\ddot{\underline{h}} = \underline{0}$ implemented in Figure 3.2 is marginally stable and numerical error leads to drift in h. To keep constraint errors small, Baumgarte stabilization [1] is implemented.

Baumgarte stabilization replaces the differential equation $\ddot{\underline{h}} = 0$ by:

$$\ddot{\underline{h}} + 2\cdot\alpha_{stab}\beta_{stab}\dot{\underline{h}} + \alpha^2\underline{h} = 0. \tag{3.3}$$

In Equation (3.3), α_{stab} and β_{stab} are stabilization parameters chosen such that the error dynamics converge to zero. With Baumgarte stabilization, the constraint equations at the acceleration level become:

$$\underline{W}^T(\underline{q},t)\ddot{\underline{q}} + \underline{\bar{w}}_{stab}(\underline{q},\dot{\underline{q}},t) = \underline{0}, \tag{3.4}$$

where:

$$\underline{\bar{w}}_{stab}(\underline{q},\dot{\underline{q}},t) = \underline{\bar{w}}(\underline{q},\dot{\underline{q}},t) + 2\cdot\alpha_{stab}\beta_{stab}\left(\underline{W}^T(\underline{q},t)\right) + \alpha_{stab}^2\cdot\underline{h}(\underline{q},t) \tag{3.5}$$

and $\underline{\bar{w}}_{stab}$ replaces $\underline{\bar{w}}$ in Figure 3.2. Figure 3.4 illustrates the time evolution of the constraint error when the dynamics of closed-loop VM are augmented with Baumgarte stabilization terms. These dynamics are implemented in the experimental haptic system.

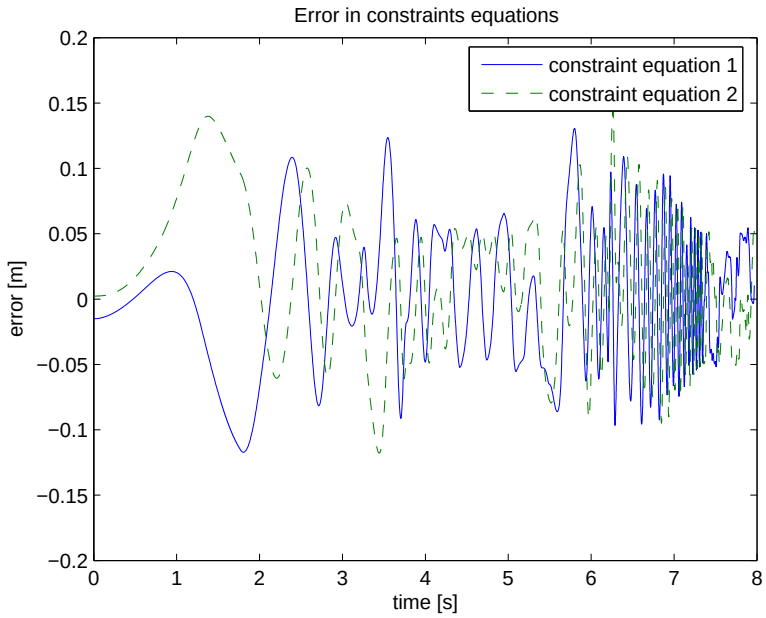


Figure 3.3: Time evolution of the loop closure constraint equation error without constraint stabilization.

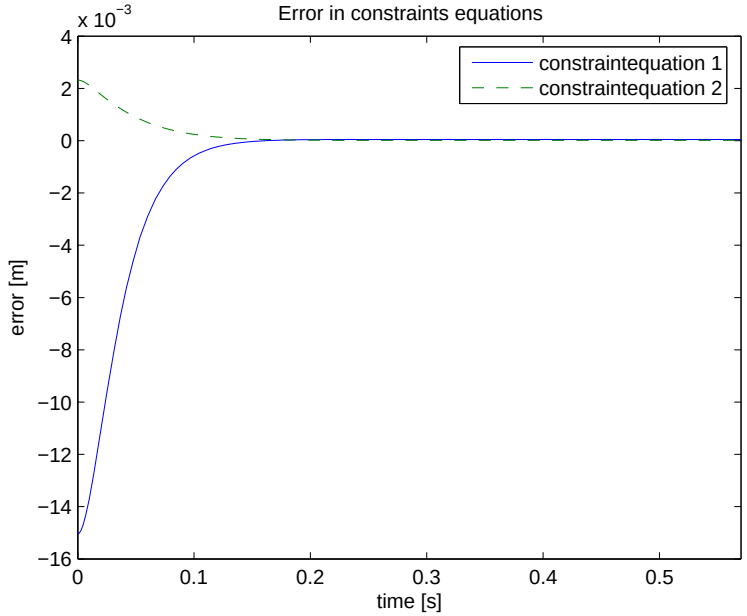


Figure 3.4: Time evolution loop closure constraint equation error with Baumgarte constraint stabilization.

The final dynamic structure implemented in the haptics block in Figure 3.2 includes Baumgarte stabilization terms and is shown in Figure 3.5.

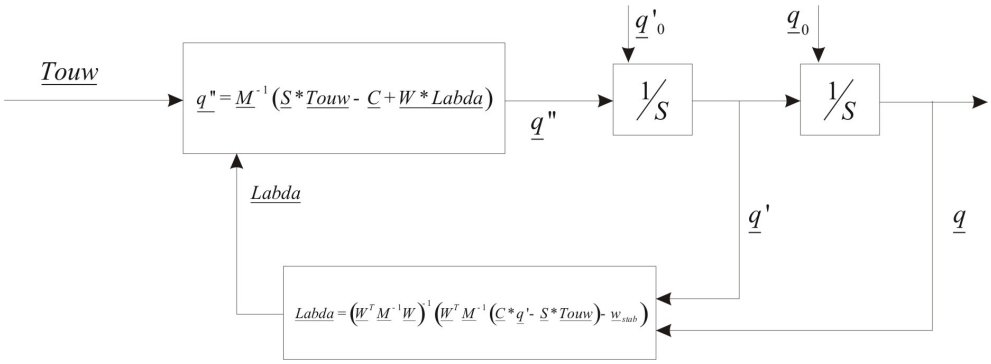


Figure 3.5: The dynamics of the closed loop VM with constraint stabilization, as implemented in the experimental setup

Chapter 4

Control

4.1 Overview of the control structure

Haptic applications enable users to touch and manipulate virtual objects. Currently, users can manipulate single objects and serial linkages using the planar haptic setup. To enable users to operate virtual linkages with closed loops, an understanding of the haptic control loops that render serial linkage manipulations is required. The following paragraphs explain these controllers and the forces that they generate in order to give users a realistic perception of the VM.

4.2 Control structure for serial linkages

This paragraph overviews the global control structure for serial linkages. Sunsequent paragraphs detail the structure of each component block. The haptic setup comprises four blocks: two haptic rendering blocks; one simulation engine block; and the haptic interface block. In this report, the audio-visual rendering blocks introduced in Chapter 1 are lumped into the simulation engine. Furthermore, the haptic simulation block (HS) represents the simulation engine and the controller block represents the haptic rendering. Haptic interface and haptic device (HD) are used interchangeably.

The block representation of the experimental setup is shown in Figure 4.1. The interactions between blocks are via forces and positions (including velocities). Force signals (F) are bidirectional whereas position signals (x) are unidirectional. Signal subscripts represent the two blocks connected by a signal. The blocks in Figure 4.1 are:

1. Haptic device: this block comprises the hardware which enables users to interact with and touch the VM;
2. Haptic simulation: this block computes the VM dynamics;
3. Four channel teleoperation controller [15]: this block connects the HD and the HS blocks;
4. Topology controller: this block connects the HD and the HS.

4.2.1 Control of the haptic device

As shown in Figure 4.1, the input to the HD block is $F_{DF} = \begin{bmatrix} \underline{F}_h & \underline{F}_{drift} & \underline{F}_{env} & \underline{T} \end{bmatrix}^T$. The computation of these forces is detailed in Sections 4.2.2, 4.2.3 and 4.2.4, and only the outputs of each block are presented herein. The outputs of the HD block are the position of the user's hand expressed in the world coordinate system, $\underline{x}_D$, and the user-applied (hand) force, $\underline{F}_h$. The hardware in the HD block has 3DOFs allowing for planar translation and unlimited rotation about a vertical axis, and is shown in Figure 4.2. The endpoints of the two pantographs that comprise the HD move in parallel planes and are

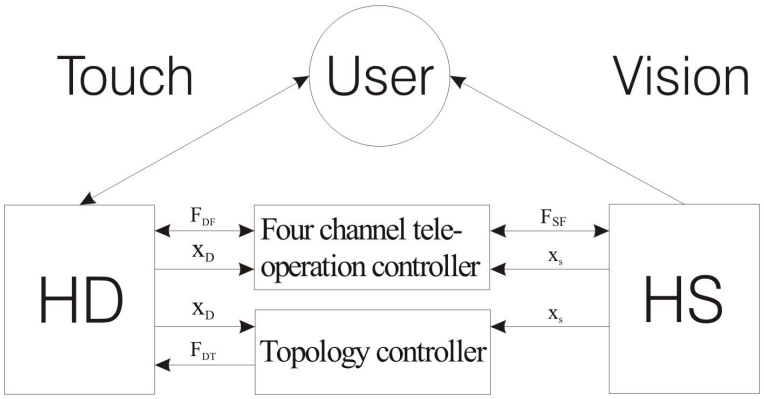


Figure 4.1: Block representation of the experimental haptic setup.

coupled by means of a bar connected to the interface handle. The connecting bar forms a crank that allows the handle to rotate unhindered. Each pantograph is driven by two DC motors located at the base joints. These motors are the actuators of the hardware mechanism. For control purposes, the motors are considered torque sources. Each of the four base joint angles is measured by a digital optical encoder with a resolution of 0.09 degrees. These encoders are the position sensors of the hardware mechanism.

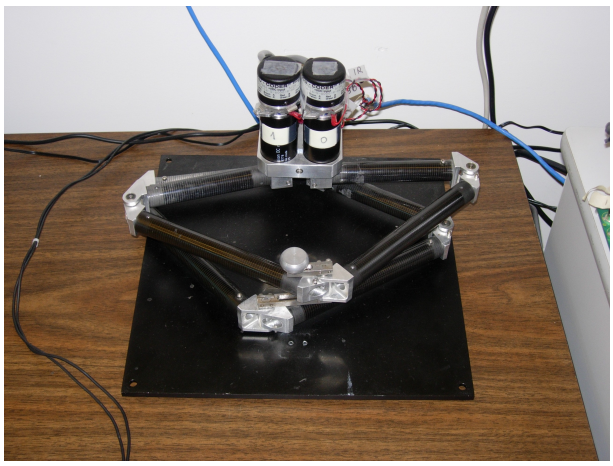


Figure 4.2: Twin pantograph planar haptic interface used in the experimental setup.

Velocities, accelerations and forces are not directly measurable. An observer block computes them using joint angle measurements and applied motor torques [25]. Given an accurate dynamic model, as well as measured joint angles and applied actuator forces, the system states (angular velocity) and unknown external disturbances (hand force applied to the hardware on the position $\underline{x}_D$) are observed and computed via a simple Nicosia observer block (for further details see [18]. The hardware provides the operator with a means of interacting with the VE. The feel of the dynamics of a VM in free motion is rendered to users via an impedance controller. The goal of the impedance controller is to shape the device dynamics to match any desired dynamics. This greatly simplifies the development of the teleoperation controller (presented in Section 4.2.3) [25].

The output of the impedance controller depends on two dynamic systems, namely the desired VM dynamics and the dynamics of the hardware. The VM dynamics are given by:

$$\underline{\Lambda}_h \ddot{\underline{x}}_h + \underline{B}_d \dot{\underline{x}}_h + \underline{K}_d \underline{x}_h = \underline{F}_{HD}, \tag{4.1}$$

where:

$$\underline{\mathbf{x}}_{\mathbf{h}} = \begin{pmatrix} x_h \\ y_h \\ \theta_h \end{pmatrix} \tag{4.2}$$

and:

$$\underline{\mathbf{F}}_{\mathbf{i}} = \begin{pmatrix} F_{h,x} + F_{drift,x} + F_{env,x} + F_{T,x} \\ F_{h,y} + F_{drift,y} + F_{env,y} + F_{T,y} \\ T_{h,z} + T_{drift,z} + T_{env,z} + T_{T,z} \end{pmatrix} \tag{4.3}$$

In Equation (4.1), $\underline{\Lambda}_{h,3x3}$, $\underline{B}_{d,3x3}$ and $\underline{K}_{d,3x3}$ are the desired inertia, damping and stiffness of the VM in operational space, respectively; $\ddot{\underline{x}}_h$ and $\dot{\underline{x}}_h$ are the desired body acceleration and velocity of the hardware (and of the user's hand) at the user-selected operational point, respectively. The VM dynamics are derived in joint space. Hence, they depend only on the joint angles and on the torques applied at the joints. In operational space, matrices and forces are represented at the user's hand, and the hand position is described in absolute coordinates. In other words, the desired VM dynamics are known at the user's hand. The operational space inertia of the VM, $\underline{\Lambda}_h$ [14] is derived in Chapter 5. Equation (4.1) is equivalent to:

$$\ddot{\underline{x}}_h = \underline{\Lambda}_h^{-1} \left(\underline{F}_{HD} - \underline{B}_d \dot{\underline{x}}_h - \underline{K}_d \underline{x}_h \right) \tag{4.4}$$

In Equation (4.4), the body acceleration of the user's hand at the user-selected operational point $\ddot{\underline{x}}_h$ is zero along a direction of motion resisted by the joints of the VM [4]. Physically, the structure of the VM restricts user's motion along such a direction. Numerically, $\underline{\Lambda}_h$ becomes infinite and $\underline{\Lambda}_h^{-1}$ becomes zero along directions opposed by the virtual joints. The virtual joints prevent user's motion along the singular directions of $\underline{\Lambda}_h^{-1}$ [4]. The hardware dynamics are given by:

$$\underline{M}_d \ddot{\underline{x}}_h + \underline{C}_d \, \dot{\underline{x}}_h = \underline{F}_h + \underline{u}, \tag{4.5}$$

where $\underline{M}_d$ and $\underline{C}_d$ are the inertia and the Christoffel matrices of the HD block's hardware, and $\underline{u}$ is the control signal. From Equations (4.4) and (4.5), the impedance control law for the HD is obtained as:

$$\underline{u} = \left(\underline{M}_d \underline{\Lambda}^{-1} - \underline{I} \right) \underline{F}_h + \underline{M}_d \underline{\Lambda}^{-1} \underline{F}_{HD} + \left(\underline{C}_d - \underline{M}_d \underline{\Lambda}^{-1} \underline{B}_d \right) \dot{\underline{x}}_h - \underline{M}_d \Lambda^{-1} \underline{K}_d \underline{x}_h. \tag{4.6}$$

The control signal in Equation (4.6) applies the contact, Coriolis, centripetal and joint constraint forces to the hardware by commanding the hardware acceleration to follow the acceleration of the VM at the user-selected operational point. Since the VM acceleration along the directions of motion constrained by the virtual joints is zero, Equation (4.6) controls to zero the hardware acceleration along these directions. However, numerical drift and limited hardware stiffness prevent zero acceleration control form effectively restricting users' motion along the directions resisted by the virtual joints. The impedance controller is unsuitable for rendering to users motion constraints due to the virtual joints. The schematic of the control structure in the HD block is depicted in Figure 4.3.

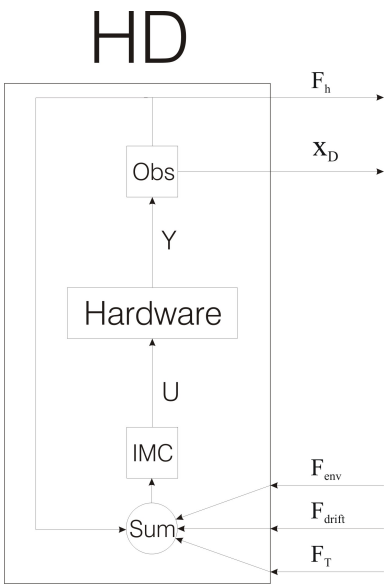


Figure 4.3: Schematic overview of the control structure in the HD block.

4.2.2 Haptic simulation

F_{SF} shown in Figure 4.1 includes three forces, $\underline{F}_h$, $\underline{F}_{drift}$ and $\underline{F}_{env}$ are three different independent force signals collected in one $\underline{F}_{SF}$ signal. $\underline{F}_{drift}$ and $\underline{F}_h$ are inputs to the HS block. $\underline{F}_{drift}$ is detailed in Section 4.2.3. The outputs of the HS block are $\underline{x}_S$ and $\underline{F}_{env}$, where $\underline{F}_{env}$ is the interaction force between the VM and other objects in the VE. The VM dynamics are computed in the HS block. These dynamics are the desired dynamics, which the user wants to feel. The matrices in Equation (4.6) and the directions of motion resisted by the virtual joints are computed in this block as well. As introduced in Section 1.3.4, the HS block comprises a haptics (H) block and a graphics (G) block, connected through a graphics controller. The separation is done because of computation time. The control structure in the HS block is shown in Figure 4.4.

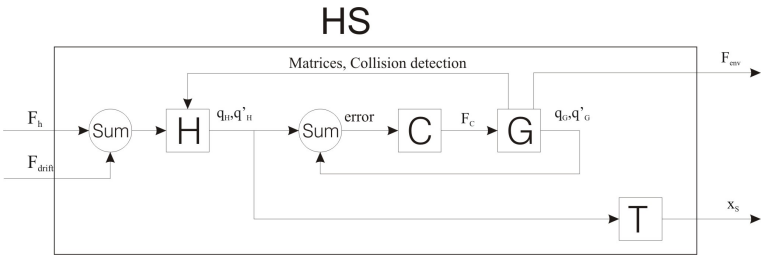


Figure 4.4: Schematic overview of the control structure in the HS block

Figure 4.4 illustrates that the haptics and the graphics blocks are not independent. An asychronous controller connects the two simulations. The haptics block runs at the same sample time as the HD block and connects the graphics and HD blocks. The graphics block runs its own VE simulation and provides visual feedback to the user using the Vortex^{tm} simulation engine. The time expensive computations are performed in the graphics block because this block runs at lower (variable) sample time than the haptics block. The graphics block computes the system matrices and performs collision detection. The matrices can be computed slower than the sampling rate of the haptics block because humans are not very sensitive to the force direction and magnitude. here). Rather, the time variation of the force is important [15]. For example, the inertia matrix is dependent on the system configuration. The configuration is obtained

via double integration of the acceleration, i.e., the second order VM system acts as a second order filter. Considering that the output amplitude is small, it follows that changes in the inertia matrix have little influence on the acceleration (the output of the VM dynamics). The second order filter behaviour of the VM dynamics implies that the system matrices can be computed at a slower rate. Collision detection is performed in the graphics block because it takes to much computation time to run in the haptics block. The haptics block computes the VM dynamics, including the unilateral contact forces ($\underline{F}_{env}$) when the graphics block informs the haptic block that collisions exist. The output of the haptics block ($\underline{x}_S$) is the reference signal for the graphics block. The error between the hand position on the VM in the haptic and graphic blocks is fed to the PD graphics controller (C) which drives this error to zero. The generalized coordinates are transformed into operational space coordinates in the transformation block (T).

4.2.3 Four channel teleoperation controller

The four channel teleoperation controller block has four inputs, $\underline{x}_S$, $\underline{x}_D$, $\underline{F}_{env}$ and $\underline{F}_h$. Its outputs are $\underline{F}_{env}$, $\underline{F}_h$ and $\underline{F}_{drift}$ signals. $\underline{F}_{drift}$ is the force output of the PD controller integral to the teleoperation controller. The four channel teleoperation controller feedforwards the environment force $\underline{F}_{env}$ to the HD block and the user-applied force $\underline{F}_h$ to the HS block. Its main purpose is to enable users to feel the contacts of the VM with the VE. In addition, it prevents the drift between the HD and HS blocks. The drift is due to numerical errors and (computational and communication) delays in the haptic system. Via its two position channels, the four channel teleoperation controller computes the error between the HS and HD blocks. The error is fed into a PD controller and $\underline{F}_{drift}$ is fed back to both blocks, as shown in Figure 4.5.

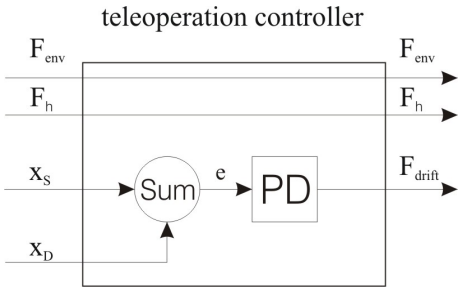


Figure 4.5: Schematic overview of the control structure in the four channel teleoperation controller block.

$\underline{F}_{drift}$ needs to be small compared to $\underline{F}_{env}$ and $\underline{F}_h$, such that the user feels the VM interactions with the VE and not $\underline{F}_{drift}$ due to numerical errors, as intuitively illustrated in Figure 4.6. Therefore, the PD controller in the teleoperation architecture is weak.

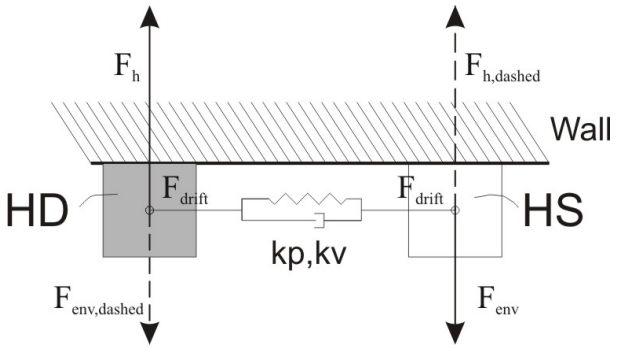


Figure 4.6: Physical interpretation of the teleoperation controller at collision.

4.2.4 Topology controller

The inputs to the topology controller block are $\underline{x}_S$ and $\underline{x}_D$. The output of the HD block is $\underline{F}_T$, which is the force output of the topology controller. The topology controller is implemented in the control structure because the impedance controller cannot restrict user's motion along the directions resisted by the virtual joints. The topology controller is a PD controller much stiffer than the position channels of the teleoperation controller, such that users perceive the joint constraints of the VM realistically. The topology controller is active only when the user manipulates the VM along a direction resisted by the virtual joints, i.e., along a singular direction of $\underline{\Lambda}_h^{-1}$, as illustrated in Figure 4.7.

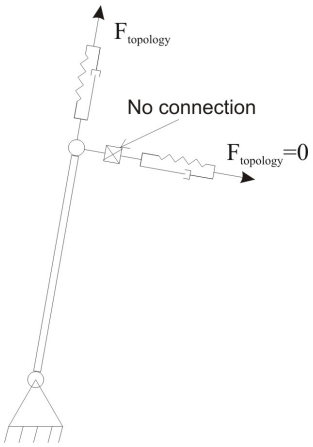


Figure 4.7: Physical interpretation of the topology controller.

4.2.5 Summary

The experimental haptic setup uses the overviewed control structure to enable users to interact with planar VEs. The structure comprises the following control loops:

1. Impedance controller. This controller renders the inertia of the VM via shaping the impedance of the haptic device to match the impedance of the VM;

2. Four channel teleoperation controller. This controller renders the contacts of the VM with the VE. The teleoperation controller feedforwards hand and environment forces between the haptic device (user) and the haptics simulation via the two force channels. It eliminates the drift between them via the two position channels;

3. Graphics controller. This controller transmits the motion of the user's hand to the graphics simulation and drives the VM in the graphics simulation to the VM in the haptic one;

4. Topology controller. This controller renders the resistance of the virtual joints via penalizing user's motion along the singular directions of $\underline{\Lambda}_h^{-1}$.

In this structure, the impedance and the topology controllers depend on the inertial properties of the VM. In particular, these controllers require $\underline{\Lambda}_h^{-1}$ and its singular directions. To enable users to realistically operate VM with closed loops in addition to single objects and serial linkages, a method for computing the inverse of the operational space inertia $\underline{\Lambda}_h^{-1}$ and its singular directions is proposed in the following chapter.

Chapter 5

Haptic manipulation of VM with closed loops

As discussed above, realistic haptic manipulations of VM with closed loops require the control structure to render the VM inertia and joint constraints. In contrast to $\underline{\Lambda}_h$ for serial linkages with dynamics in independent coordinates, $\underline{\Lambda}_h$ for mechanisms with closed loops and dynamics in dependent coordinates incorporates the loop closure constraints. To highlight the representation of the loop closure constraints in $\underline{\Lambda}_h$, the derivation of $\underline{\Lambda}_h$ for closed loop linkages is preceded by an overview of the computation of $\underline{\Lambda}_h^{-1}$, for open loop linkages in the following section.

5.1 Open loop mechanisms

The derivations in this section follow [14]. Assuming small velocities (i.e., ignoring Coriolis and centripetal effects), the dynamics of serial linkages in independent coordinates are obtained from Equation (2.3) as:

$$\underline{D}\ddot{\underline{q}} = \underline{J}_h^T \underline{F}_h \qquad \Rightarrow \qquad \ddot{\underline{q}} = \underline{D}^{-1} \underline{J}_h^T \underline{F}_h. \tag{5.1}$$

In operational space, these dynamics are:

$$\underline{\Lambda}_h \ddot{\underline{x}}_h = \underline{F}_h. \tag{5.2}$$

After transformation of the generalized acceleration into body acceleration at the user-selected operational space point via:

$$\ddot{\underline{x}}_h = \underline{J}_h \ddot{\underline{q}} = \underline{\Lambda}_h^{-1} \underline{F}_h, \tag{5.3}$$

the inverse of the operational space inertia matrix Λ_h^{-1} can be derived:

$$\underline{\Lambda}_h^{-1} = \underline{J}_h^T \underline{D}^{-1} \underline{J}_h^T \tag{5.4}$$

5.2 Closed loop mechanisms

The dynamics of closed loop mechanisms in dependent coordinates includes the vector of Lagrange multipliers which enforce the loop closure constraints:

$$\underline{D}\underline{\ddot{q}} = \underline{J}_h^T \underline{F}_h + \underline{W}^T \underline{\lambda} \tag{5.5}$$

From Equation (5.5), the generalized acceleration of the closed loop mechanism can be computed via:

$$\underline{\ddot{q}} = \underline{D} \left(\underline{J}_h^T \underline{F}_h + \underline{W}^T \underline{\lambda} \right). \tag{5.6}$$

The body acceleration of the user's hand at the user-selected operational point is:

$$\underline{\ddot{x}}_h = \underline{J}_h \underline{\ddot{q}} = \underline{\Lambda}_h^{-1} \underline{F}_h. \tag{5.7}$$

After substitution of Equation (5.6) into Equation (5.7), $\underline{\Lambda}_h^{-1}$ becomes:

$$\underline{\Lambda}_h^{-1} \underline{F}_h = \underline{J}_h \underline{D}^{-1} (\underline{J}_h^T \underline{F}_h + \underline{W}^T \underline{\lambda}) \tag{5.8}$$

The unknowns in Equation (5.8) are $\underline{\Lambda}_h^{-1}$ and $\underline{\lambda}$ Therefore, the constraint equations are used to compute the Lagrange multipliers:

$$\underline{\lambda} = - \left((\underline{W D}^{-1} \underline{W}^T)^{-1} \underline{W D}^{-1} \underline{J}_h^T \right) \underline{F}_h. \tag{5.9}$$

Substitution of Equation (5.9) into Equation (5.8) gives:

$$\underline{J}_h \underline{D}^{-1} \left(\underline{J}_h^T - \underline{W}^T - \left((\underline{W D}^{-1} \underline{W}^T)^{-1} \underline{W D}^{-1} \underline{J}_h^T \right) \right) \underline{F}_h = \underline{\Lambda}_h^{-1} \underline{F}_h, \tag{5.10}$$

and allows the computation of the inverse of the operational space inertia of the closed loop VM at the user-selected operational point:

$$\underline{\Lambda}_h^{-1} = \underline{J}_h \underline{D}^{-1} \left(\underline{J}_h^T - \underline{W}^T - \left((\underline{W D}^{-1} \underline{W}^T)^{-1} \underline{W D}^{-1} \underline{J}_h^T \right) \right) \tag{5.11}$$

The singular directions of $\underline{\Lambda}$ are derived via singular value decomposition. As proposed in [4], both $\underline{\Lambda}$ and its singular directions are computed in the graphics simulation and are sent to the haptic simulation at each update. This saves computation time and allows the haptic simulation to run at the $512Hz$ sampling frequency of the control loops.

Chapter 6

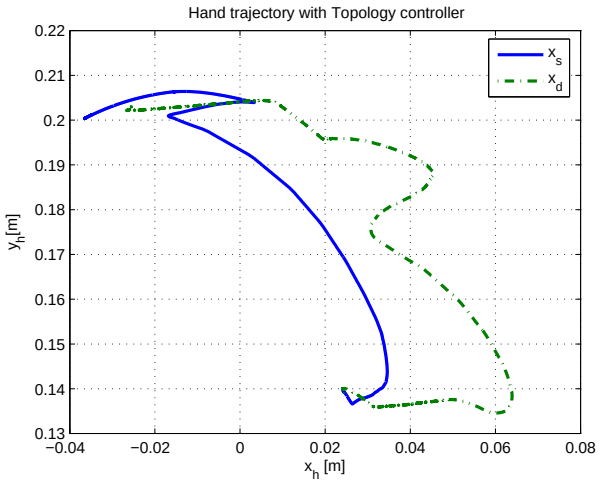
Experimental results

User's perception of loop closure constraints during the manipulation of VM with closed loops was investigated through a controlled experiment performed using the experimental setup at the Assistive Robotics and Mechatronics Systems (ARMS) Laboratory in the Department of Mechanical Engineering at the University of Victoria. In the experiment, a controlled user-applied force is employed. This means that $\underline{F}_h$ is applied by a block added to the Simulink diagram rather than by hand. The controlled user-applied force makes the experiment repeatable and allows results of successive experiments to be compared. The "user" operates the VM depicted in Figure 3.1 from the COM of the second bar. The VM dimensions are given in Table 6.1.

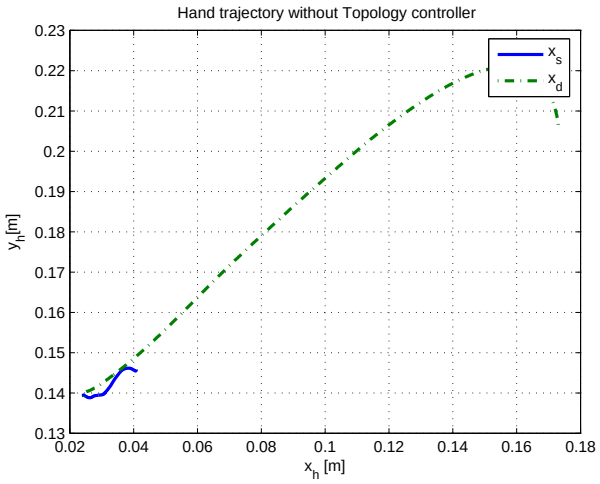
Table 6.1: Dimensions of the planar VM operated by the "user" in the controlled experiments.

Link length [cm]	Link mass [kg]	Link inertia [kg(cm)^2]
$l_1 = 4.2$	m_1=1	I_1=2.1
$l_2 = 4.2$	m_2=1	I_2=2.1
$l_3 = 1.84$	m_3=0.5	I_3=1.05
$l_4 = 5.64$	m_4=0.5	I_4=1.05

Two experiments are performed. In the first experiment, the topology controller is active. In other words, the impedance controller renders the VM inertia to the "user"'s hand and the stiffness of the topology controller represents the joint constraints to the "user"'s hand. In the second experiment, the topology controller is inactive, i.e., the impedance controller is used to apply both the VM inertia and the joint constraints to the "user"'s hand. In both experiments, a constant force and torque $F_h = [-0.5N\ 0N\ -0.0025Nm]^T$ represent the user. Furthermore, the linkage is initially at rest, in the configuration space position shown in Figure 3.1, $q_0 = \begin{pmatrix}\pi/4 & -\pi/2 & -4\pi/10 & \pi/2\end{pmatrix}^T$. The experimental device trajectory (HD) and the simulated hand trajectory (HS) are plotted in 6.3(a) for the first experiment, and in 6.3(b) for the second experiment.
In Figure 6.2 and Figure 6.3 the difference of HD and HS in time are depicted, the figure shows that the error between HS is smaller with topology controller than without.
It has to be noted that the experiment with inactive topology controller is stopped before the haptic device reaches the hard constraints imposed by the workspace limits of the haptic interface.

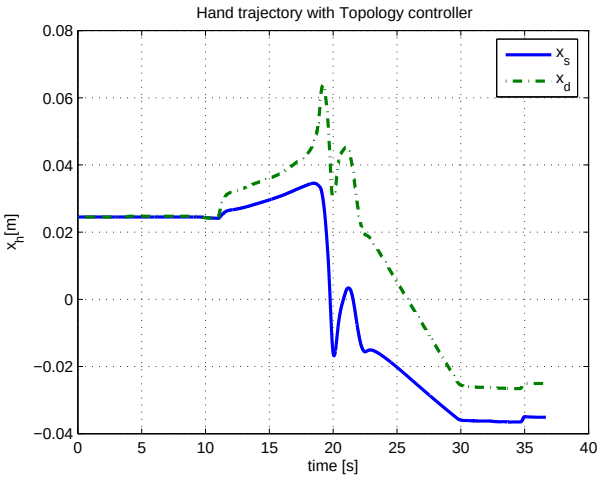


(a) Active topology controller.

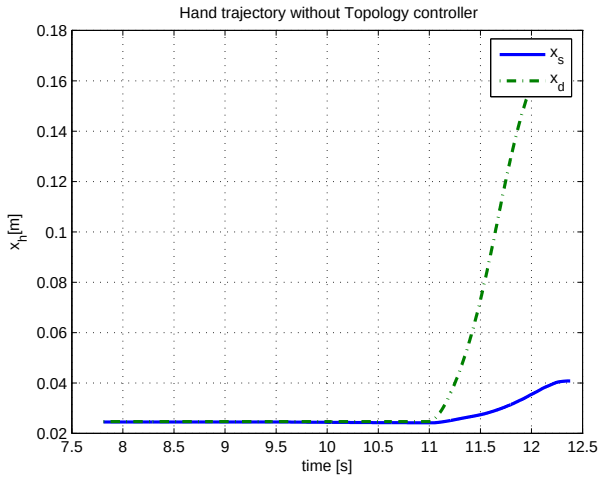


(b) Inactive topology controller.

Figure 6.1: Experimental hand trajectories obtained when a constant force and torque are applied to the COM of the second link of the VM shown in Figure 3.1

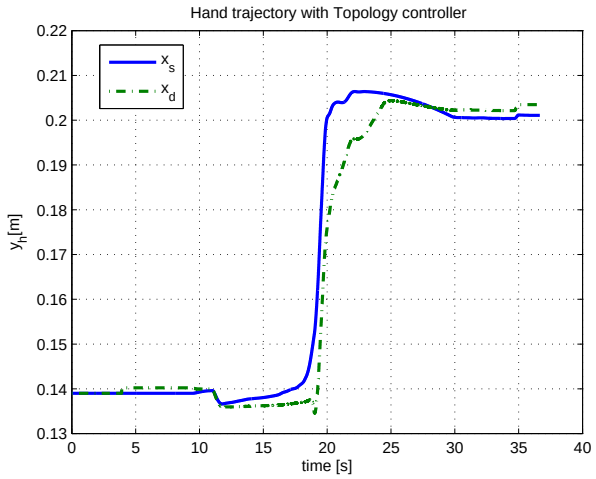


(a) Active topology controller.

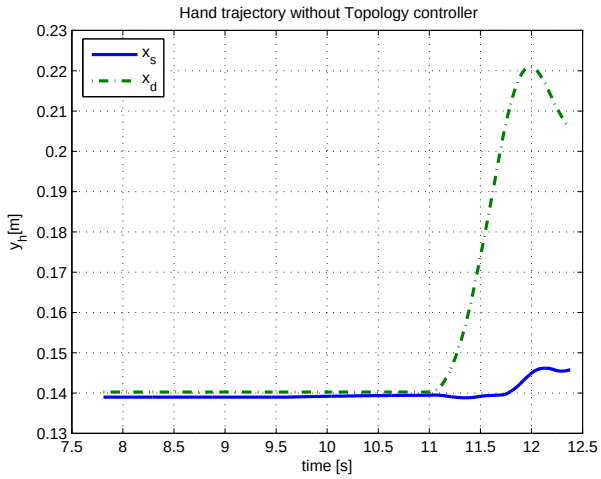


(b) Inactive topology controller.

Figure 6.2: x-position of trajectories



(a) Active topology controller.



(b) Inactive topology controller.

Figure 6.3: y-position of trajectories

Chapter 7

Conclusion

This report has proposed an approach enabling users to manipulate virtual linkages with closed loops. In this approach, linkage dynamics are simulated in configuration space and loop closure constraints are maintained via Lagrange multipliers augmented with Baumgarte stabilization terms. The virtual dynamics are rendered to users in Cartesian space. The varying linkage inertia is applied via impedance control in operational space, and the joint contraints are enforced via stiffness control along directions orthogonal to operational space. The ability to restrict user' s motion as required by the virtual closed loops has been validated through an experiment performed using a planar haptic interaction system. Future work will investigate the simultaneous manipulation of a virtual linkage by multiple users, and techniques for automatically selecting the gains of the stiffness controller such that users perceive equally stiff contacts and joint constraints.

Appendix A

Matrices used in the implementation

This Appendix details the matrices used in the Lagrange equations of motion of the closed loop VM implemented in the haptic interaction system. See Figure 3.1 for details. The VM is described in generalized coordinates $\underline{q} = \begin{pmatrix}\theta_1 & \theta_2 & \theta_3 & \theta_4\end{pmatrix}^T$. In the sequel, $c_i = \cos(q_i)$ and $s_i = \sin(q_i)$. For further details of the method used to derive the dynamics of the closed loop VM, see [26].
After attaching a frame to each body, all frames can be described with respect to a fixed frame $\vec{e}_0$:

$$\vec{e}^{\,1} = A^{10}(q_1)\vec{e}^{\,0} = \begin{bmatrix} c_1 & s_1 \\ -s_1 & c_1 \end{bmatrix} \vec{e}^{\,0} \tag{A.1}$$

$$\vec{e}^{\,2} = A^{21}(q_2)\vec{e}^{\,1} = \begin{bmatrix} c_2 & s_2 \\ -s_2 & c_2 \end{bmatrix} \vec{e}^{\,1} \tag{A.2}$$

$$\vec{e}^{\,3} = A^{30}(q_3)\vec{e}^{\,0} = \begin{bmatrix} c_3 & s_3 \\ -s_3 & c_3 \end{bmatrix} \vec{e}^{\,0} \tag{A.3}$$

$$\vec{e}^{\,4} = A^{43}(q_4)\vec{e}^{\,3} = \begin{bmatrix} c_4 & s_4 \\ -s_4 & c_4 \end{bmatrix} \vec{e}^{\,3} \tag{A.4}$$

See [26] for details of the used vectors:

$$\begin{array}{lll} \vec{c}_0 = [x_1 \quad y_1]\vec{e}^{0} = \vec{0} & & \text{Joint } Q_1 \\ \vec{c}_1 = [x_2 \quad y_2]\vec{e}^{1} = \vec{0} & & \text{Joint } Q_2 \\ \vec{c}_2 = [x_3 \quad y_3]\vec{e}^{2} = \vec{0} & & \text{Joint } Q_3 \\ \vec{c}_3 = [x_4 \quad y_4]\vec{e}^{3} = \vec{0} & & \text{Joint } Q_4 \end{array}$$

$$\vec{r}_{cm,2} + \vec{b}_{25} = \vec{r}_{cm,4} + \vec{b}_{45} \qquad , with\ \vec{b}_{\#\ ofbody,\#\ of\ Joint} \qquad \text{CutJoint } Q_5 \tag{A.5}$$

$$\vec{r}_{cm,1} = \begin{bmatrix} f_1 L_1 c_1 \\ f_1 L_1 s_1 \end{bmatrix} \vec{e}^{\,0}$$

$$\vec{r}_{cm,2} = \begin{bmatrix} L_1 c_1 + (1-f_2) L_2 c_{1+2} \\ L_1 s_1 + (1-f_2) L_2 s_{1+2} \end{bmatrix} \vec{e}^{\,0}$$

$$\vec{r}_{cm,3} = \begin{bmatrix} f_3 L_3 c_3 \\ f_3 L_3 s_3 \end{bmatrix} \vec{e}^{\,0}$$

$$\vec{r}_{cm,4} = \begin{bmatrix} L_3c_3 + (1-f_4)L_4c_{3+4} \\ L_3s_3 + (1-f_4)L_4s_{3+4} \end{bmatrix} \vec{e}^{\,0}$$

$$\vec{b}_{25} = \begin{bmatrix} (1-f_2)L_2c_{1+2} \\ (1-f_2)L_2s_{1+2} \end{bmatrix} \vec{e}^{\,0}$$

$$\vec{b}_{45} = \begin{bmatrix} (1-f_4)L_4c_{3+4} \\ (1-f_4)L_4s_{3+4} \end{bmatrix} \vec{e}^{\,0}$$

From the equations above, the constraint equations are as follows:

$$\underline{h} = \begin{bmatrix} L_1c_1 + L_2c_{1+2} - L_3c_3 - L_4c_{3+4} \\ L_1s_1 + L_2s_{1+2} - L_3s_3 - L_4s_{3+4} \end{bmatrix}$$

The Jacobians are derived to calculate $\underline{\Lambda}$ (see Equation (2.9):

$$\underline{J}_{cm,1} = \begin{bmatrix} -f_1L_1s_1 & 0 & 0 & 0 \\ f_1L_1c_1 & 0 & 0 & 0 \end{bmatrix}$$

$$\underline{J}_{cm,2} = \begin{bmatrix} -L_1s_1 - f_2L_2s_{1+2} & -f_2L_2s_{1+2} & 0 & 0 \\ L_1c_1 + f_2L_2c_{1+2} & f_2L_2c_{1+2} & 0 & 0 \end{bmatrix}$$

$$\underline{J}_{cm,3} = \begin{bmatrix} 0 & 0 & -f_3L_3s_3 & 0 \\ 0 & 0 & f_3L_3c_3 & 0 \end{bmatrix}$$

$$\underline{J}_{cm,4} = \begin{bmatrix} 0 & 0 & -L_3s_3 - f_4L_4s_{3+4} & -f_4L_4s_{3+4} \\ 0 & 0 & L_3c_3 + f_4L_4c_{3+4} & f_4L_4c_{3+4} \end{bmatrix}$$

$$\underline{\omega} = \begin{bmatrix} \dot{q}_1 \\ \dot{q}_1 + \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_3 + \dot{q}_4 \end{bmatrix}$$

$$\underline{K} = 1/2(\underline{\omega}^T \underline{I}_{cm} \underline{\omega}) = 1/2(J_1\dot{q}_1^2 + J_2(\dot{q}_1 + \dot{q}_2)^2 + J_3\dot{q}_3^2 + J_4(\dot{q}_3 + \dot{q}_4)^2) \tag{A.6}$$

$$\underline{I} = \frac{\partial\left(\frac{\partial K}{\partial \underline{\dot{q}}}\right)}{\partial t} = \begin{bmatrix} J_1 + J_2 & J_2 & 0 & 0 \\ J_2 & J_2 & 0 & 0 \\ 0 & 0 & J_3 + J_4 & J_4 \\ 0 & 0 & J_4 & J_4 \end{bmatrix}$$

The $\underline{D}$ matrix is given by:

$$\underline{D} = \underline{\Lambda} + \underline{I} = \begin{bmatrix} D_{11} & D_{12} & 0 & 0 \\ D_{12} & D_{22} & 0 & 0 \\ 0 & 0 & D_{33} & D_{34} \\ 0 & 0 & D_{43} & D_{44} \end{bmatrix},$$

with

$$\begin{aligned} D_{11} &= m_1 f_1^2 L_1^2 + m_2 L_1^2 + 2m_2 L_1 f_2 L_2 c_2 + m_2 f_2^2 L_2^2 + J_1 + J_2 \\ D_{12} &= m_2 L_1 f_2 L_2 c(_2) + m_2 f_2^2 L_2^2 + J_2 \\ D_{21} &= m_2 L_1 f_2 L_2 c_2 + m_2 f_2^2 L_2^2 + J_2 \\ D_{22} &= m_2 f_2^2 L_2^2 + J_2 \\ D_{33} &= m_3 f_3^2 L_3^2 + m_4 L_3^2 + 2m_4 L_3 f_4 L_4 c_4 + m_4 f_4^2 L_4^2 + J_3 + J_4 \\ D_{34} &= m_4 L_3 f_4 L_4 c_4 + m_4 f_4^2 L_4^2 + J_4 \\ D_{43} &= m_4 L_3 f_4 L_4 c_4 + m_4 f_4^2 L_4^2 + J_4 \\ D_{44} &= m_4 f_4^2 L_4^2 + J_4 \end{aligned} \tag{A.7}$$

The $\underline{C}$ matrix is given by (see Equation (2.11):

$$\underline{C} = \begin{bmatrix} -m_2 L_1 f_2 L_2 s_2 \dot{q}_2 & -m_2 L_1 f_2 L_2 s_2 (\dot{q}_1 + \dot{q}_2) & 0 & 0 \\ m_2 L_1 f_2 l_2 s_2 \dot{q}_1 & 0 & 0 & 0 \\ 0 & 0 & -m_4 L_3 f_4 L_4 s_4 \dot{q}_4 & -m_4 L_3 f_4 L_4 s_4 (\dot{q}_3 + \dot{q}_4) \\ 0 & 0 & m_4 L_3 f_4 l_4 s_4 \dot{q}_3 & 0 \end{bmatrix}$$

The $\underline{W}$ matrix is given by (see Equation (2.14):

$$\underline{W} = \frac{\partial \underline{h}}{\partial \underline{q}} = \begin{bmatrix} -L_1 s_1 - L_2 s_{1+2} & -L_2 s_{1+2} & L_3 s_3 + L_4 s_{3+4} & L_4 s_{3+4} \\ L_1 c_1 L_2 c_{1+2} & L_2 c_{1+2} & -L_3 c_3 - L_4 c_{3+4} & -L_4 c_{3+4} \end{bmatrix}$$

The external force is applied on the the centre of mass of the 2^{nd} bar. The position $\underline{r}_h$ and the $\underline{J}_h$ matrix are given by (see Equation (2.18):

$$\underline{r}_h = \begin{bmatrix} L_1 c_1 + f_2 L_2 c_{1+2} \\ L_1 s_1 + f_2 L_2 s_{1+2} \\ q_1 + q_2 \end{bmatrix}$$

$$\underline{J}_h = \begin{bmatrix} -L_1 s_1 - f_2 L_2 s_{1+2} & -f_2 L_2 s_{1+2} & 0 & 0 \\ L_1 c_1 + f_2 L_2 c_{1+2} & f_2 L_2 c_{1+2} & 0 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

In Equation (3.1) and (3.1) the $\bar{\underline{w}}$ is given by:

$$\bar{\underline{w}} = \begin{bmatrix} ((-L_1 c_1 - L_2 c_{1+2})\dot{q}_1 - L_2 c_{1+2}\dot{q}_2)\dot{q}_1 + (-L_2 c_{1+2}\dot{q}_1 - L_2 c_{1+2}\dot{q}_2)\dot{q}_2 + \\ ((-L_1 s_1 - L_2 s_{1+2})\dot{q}_1 - L_2 s_{1+2}\dot{q}_2)\dot{q}_1 + (-L_2 s_{1+2}\dot{q}_1 - L_2 s_{1+2}\dot{q}_2)\dot{q}_2 + \\ \\ ((L_3 c_3 + L_4 c_{3+4})\dot{q}_3 + L_4 c_{3+4}\dot{q}_4)\dot{q}_3 + (L_4 c_{3+4}\dot{q}_3 + L_4 c_{3+4}\dot{q}_4)\dot{q}_4 \\ ((L_3 s_3 + L_4 s_{3+4})\dot{q}_3 + L_4 s_{3+4}\dot{q}_4)\dot{q}_3 + (L_4 s_{3+4}\dot{q}_3 + L_4 s_{3+4}\dot{q}_4)\dot{q}_4 \end{bmatrix}$$

Appendix B

Paper Icra 2007: Haptic Manipulation of Virtual Linkages with Closed Loops

B.1 Introduction

Haptics applications strive to provide force feedback to users interacting with realistic virtual environments. A key factor affecting the realism of the virtual environment is the complexity of the virtual objects that users can manipulate (hereafter called the virtual tools, VTs). This complexity is typically limited by the stringent speed requirements of the haptic controller. For compelling interaction within rigid virtual worlds, the haptic controller demands the virtual dynamics to be computed at guaranteed rates of the order of hundreds of Hz. Many haptics applications meet the severe constraints of the haptic controller through enabling users to touch and manipulate only single virtual objects. More complex VTs may be needed in various applications. For example, during virtual reality-based training for robot assisted minimally invasive surgery, surgeons need to manipulate VTs with motion constrained by the robotic system. Virtual CAD prototyping and animation applications may also require users to operate virtual linkages. In contrast to the haptic manipulation of single objects, realistic force interaction with virtual linkages commands the haptic rendering of the linkage dynamics, including the varying inertia and the joint constraints.

Initial haptics work has focused on linkage simulations that run at the speed of the haptic controller. Computational efficiency has been achieved via application-specific [17] and general purpose algorithms [21, 23]. Increased physical accuracy of the haptic simulation of linkages has been sought via modeling collisions in [22, 5]. The haptic rendering of the joint constraints of serial-chain linkages has been implemented via proxies with first order dynamics in [16] and via a controller acting along the constrained directions in [7]. Apart from the haptic manipulation of a closed loop virtual linkage reported in [17] and based on an application-specific simulation and haptic rendering algortihm, mostly haptic interaction with serial-chain mechanisms has been investigated to date.

The present research expends the class of VTs that users can manipulate to include passive linkages with closed loops (linkages under feedback control, i.e., active, can also be included in the proposed formulation). The focus is on enabling users to feel the varying virtual inertia and motion restrictions due to the closed loops. The inertia of VTs with closed loops is derived at an arbitrary user-selected manipulation point, and employed to compute the motions restricted by the virtual joints. Users feel the virtual inertia through impedance control. They feel the directions of constraint via stiffness control. As illustrated in Section B.6, stiffness control effectively restricts users' motion as required by the closed loops.

In the remainder of the paper, the simulation of VTs with closed loops enforced via Lagrange multipliers is discussed in Section B.2. The haptic rendering of the dynamics of VTs with closed loops to users is overviewed in Sections B.3. The derivation of the VT inertia and of the loop closure constraints at the user-selected operational point is developed in Section B.4. The efficient

implementation of the proposed approach is detailed in Section B.5. Haptic rendering of closed loops is validated through experiments in Section B.6. Concluding remarks are presented in Section B.7.

B.2 Linkage Simulation

Much multibody dynamics, robotics and computer graphics research has addressed the simulation of multibody systems with constraints, of which VTs with closed loops are a sub-class. One challenge in simulating such systems is the incorporation of constraints into the equations of motion. Existing methods fall into one of three categories [10]: the method of Lagrange multipliers, which includes the unknown constraint forces in the unconstrained dynamics; the generalized coordinate partitioning method, which projects the unconstrained dynamics on a subspace compatible with the constraints numerically; and the method of constraint embedding which projects the unconstrained dynamics on a subspace compatible with the constraints symbolically. The first two methods produce differential algebraic equations (DAEs) whose integration is unsuitable for real time performance [10]. Differentiating the constraints twice transforms the DAEs into ordinary differential equations (ODEs), but requires constraint stabilization. Constraint embedding produces ODEs that do not require stabilization, but symbolic computations currently prohibit haptic numerical performance. The numerical inefficiency of symbolic constraint embedding has been sidestepped in [11, 12] through a multi-rate architecture in which the symbolic computations are performed outside the main integration loop. This architecture has been shown to automatically generate the equations of motion of multibody systems with changing constraints, and it is envisioned that computational power increases will soon allow its use for haptic interaction.

The present work simulates VTs with closed loops through the method of Lagrange multipliers in its form common in multibody dynamics [19, 20, 10]. Specifically, the VT dynamics are derived using extended generalized coordinates [8]. These coordinates include the relative joint coordinates identified after selecting cut joints and cutting them open, as illustrated in Fig. B.1 for a VT with one closed loop.

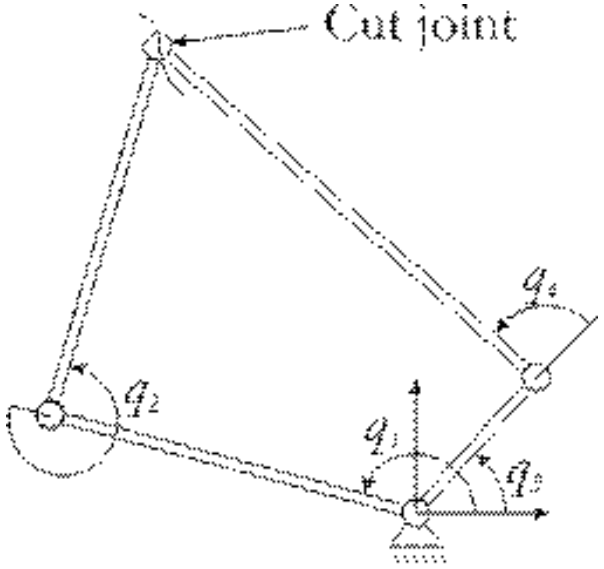


Figure B.1: Choice of configuration variables for a VT with closed loops.

In the chosen (dependent) configuration variables, the dynamics of a VT with n links, m closed loops

and c contacts are given by:

$$\begin{aligned}\mathbf{D}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{B}(\mathbf{q},\dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) &= \sum_{i=1}^{c} \mathbf{J}_i^T(\mathbf{q})\mathbf{F}_i + \mathbf{J}_h^T(\mathbf{q})\mathbf{F}_h \\ &\quad -\mathbf{W}^T(\mathbf{q})\boldsymbol{\lambda},\end{aligned} \tag{B.1}$$

where: $\mathbf{D}(\mathbf{q})_{n\times n}$ is the configuration space mass (inertia) matrix of the VT; $\mathbf{B}(\mathbf{q},\dot{\mathbf{q}})_{n\times 1}$ groups Coriolis and centripetal terms; $\mathbf{G}(\mathbf{q})_{n\times 1}$ encompasses joint flexibility and gravitational effects; $\mathbf{J}_i(\mathbf{q})_{6\times n}$ is the linkage Jacobian computed at the i-th contact; $\mathbf{F}_{i_{6\times 1}} = \begin{pmatrix}\mathbf{f}_i^T & \mathbf{0}^T\end{pmatrix}^T$ is the contact wrench (i.e., the force $\mathbf{f}_{i_{3\times 1}}$ and torque $\boldsymbol{\tau}_{i_{3\times 1}} = \mathbf{0}_{3\times 1}$) at the i-th contact; $\mathbf{J}_h(\mathbf{q})_{6\times n}$ is the VT Jacobian computed at the user's hand (hand Jacobian); $\mathbf{F}_{h_{6\times 1}}$ is the wrench applied by the user; $\mathbf{W}(\mathbf{q})_{m\times n} = \frac{\partial \mathbf{h}(\mathbf{q})}{\partial \mathbf{q}}$ is the Jacobian of the loop closure constraints; $\boldsymbol{\lambda}_{m\times 1}$ is the vector of Lagrange multipliers, i.e., unknown magnitudes of constraint forces maintaining the closed loops at the cut joints; and $\mathbf{q}_{n\times 1}$, $\dot{\mathbf{q}}_{n\times 1}$, and $\ddot{\mathbf{q}}_{n\times 1}$ are the configuration space position, velocity, and acceleration of the VT, respectively. (B.1) describes the dynamics of a passive VT. Haptic manipulation of VTs under feedback control (i.e., active) can be allowed by augmenting (B.1) with the desired feedback torques.
Integration of a DAE of index three [2] is avoided through augmenting the VT dynamics in (B.1) with acceleration constraints. For the holonomic constraints considered here, the acceleration constraints result from differentiating the loop closure constraints $\mathbf{h}(\mathbf{q})_{m\times 1} = 0$ twice to obtain:

$$\ddot{\mathbf{h}} = \mathbf{W}\ddot{\mathbf{q}} + \mathbf{w} = \mathbf{0}. \tag{B.2}$$

In (B.2):

$$\mathbf{w} = \left(\frac{\partial \mathbf{W}}{\partial t} + \frac{\partial \mathbf{W}}{\partial \mathbf{q}}\dot{\mathbf{q}}\right)\cdot\dot{\mathbf{q}}, \tag{B.3}$$

and the dependence of the various terms on $\dot{\mathbf{q}}$, $\mathbf{q}$, and time t is implied. Loop closure drift is kept small through Baumgarte stabilization [1] , i.e., through replacing (B.2) with:

$$\begin{aligned}&\ddot{\mathbf{h}} + 2\alpha\beta\dot{\mathbf{h}} + \alpha^2\mathbf{h} = 0 \Leftrightarrow \\ &\mathbf{W}\ddot{\mathbf{q}} + \mathbf{w}_{stab} = \mathbf{0}.\end{aligned} \tag{B.4}$$

In (B.4):

$$\mathbf{w}_{stab} = \mathbf{w} + 2\alpha\beta\mathbf{W} + \alpha^2\mathbf{h}, \tag{B.5}$$

and α and β are chosen such that the error dynamics converge to zero.
After augmenting the VT dynamics with the Baumgarte stabilized loop closure constraints at the acceleration level, the haptic simulation solves:

$$\begin{bmatrix}\mathbf{D} & \mathbf{W}^T \\ \mathbf{W} & \mathbf{0}\end{bmatrix}\begin{pmatrix}\ddot{\mathbf{q}} \\ \boldsymbol{\lambda}\end{pmatrix} = \begin{pmatrix}\sum_{i=1}^{c}\mathbf{J}_i^T\mathbf{F}_i + \mathbf{J}_h^T\mathbf{F}_h - \mathbf{B} - \mathbf{G} \\ -\mathbf{w}_{stab}\end{pmatrix} \tag{B.6}$$

and integrates the configuration acceleration in fixed steps equal to the time step of the force feedback loop. For non-redundant loop closure constraints, $\mathbf{W}$ is non-singular and the computational delay inccured through solving $n+m$ ODEs in the proposed approach, compared to $n-m$ ODEs in methods which derive the VT dynamics in independent coordinates, is minor for VTs with a limited number of closed loops. This cost is partly offset through avoiding the need to identify new sets of configuration variables during the simulation, as would be necessary if minimal coordinates would be used. Another advantage of enforcing the loop closure constraints through Lagrange multipliers is that techniques are available for efficiently deriving the Lagrangian dynamics of multibody systems with tree kinematic structures [9].

B.3 Haptic Rendering of Linkage Dynamics

During manipulation of VTs with closed loops, users need to feel: (i) the inertia of the VT at the selected operational point; (ii) the virtual joints when attempting to move outside the space of feasible motions and (iii) the contacts of the VT when other virtual objects restrict the VT motion. The present work employs a control architecture whereby separate controllers render inertia, joint constraints and contacts to users [7]. This control architecture, schematically depicted in Fig. B.2, includes the following controllers:

1. *Impedance controller*: renders the VT dynamics during free motion of the user-selected operational point (i.e., motion unimpeded by the joints or other objects). Based on the inverse of the VT operational space inertia $\mathbf{\Lambda}^{-1}$, this controller shapes the impedance of the device to match the VT impedance at the operational point.

2. *Joint constraint controller*: renders the resistance of the virtual joints when users try to move outside the space of feasible operational space motions. In essence, this is a stiffness controller which hinders motion along the directions $\mathbf{n}_c$ resisted by the virtual joints.

3. *Four channel teleoperation controller*: renders the VT contacts through feedforwarding the hand $\mathbf{F}_h$ and contact $\mathbf{F}_i$ forces between the haptic device (user) and the haptics simulation via the two force channels. It eliminates user drift from the operational point via the two position channels.

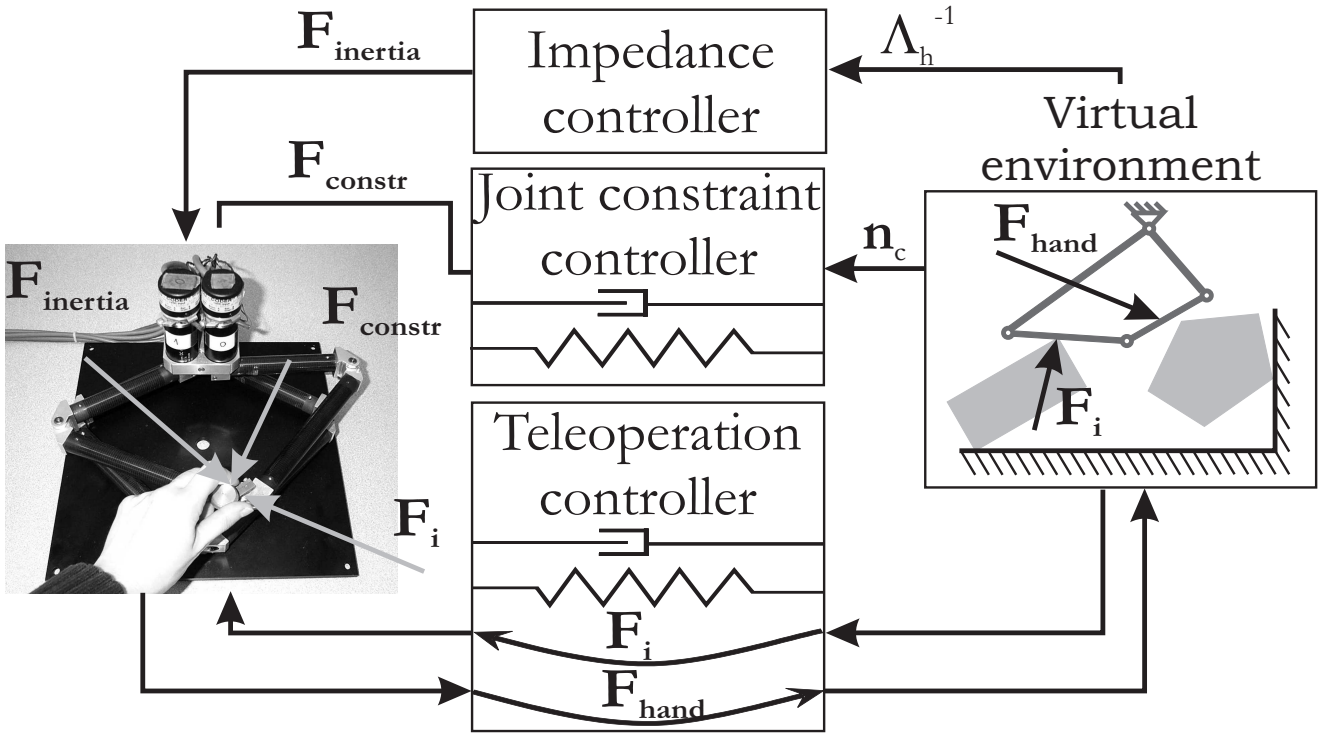


Figure B.2: Schematic of the control architecture rendering the VT dynamics.

To enable manipulations of VTs with closed loops, this architecture requires the VT inertia and the directions resisted by the cut joints at the selected operational point. A method for computing them is proposed in the following section.

B.4 Linkage inertia at the operational point

In contrast to serial-chain VTs with dynamics in independent coordinates, the dynamics of VTs with closed loops are simulated in extended generalized coordinates and the loop closure constraints are included via Lagrange multipliers. Therefore, the effect of the Lagrange multipliers on the VT operational space dynamics is needed for realistic haptic rendering of the closed loops. This effect is derived herein under the simplifying assumption that dynamic coupling effects are negligible. The validity of this assumption is verified experimentally in Section B.6.
From the dynamics of a VT with closed loops enforced through Lagrange multipliers, neglijible coupling effects and no contacts:

$$\mathbf{D}\ddot{\mathbf{q}} = \mathbf{J}_h^T\mathbf{F}_h + \mathbf{W}^T\boldsymbol{\lambda}, \tag{B.7}$$

the generalized VT acceleration can be computed via:

$$\ddot{\mathbf{q}} = \mathbf{D}^{-1}\left(\mathbf{J}_h^T\mathbf{F}_h + \mathbf{W}^T\boldsymbol{\lambda}\right). \tag{B.8}$$

The operational space acceleration of the user's hand is:

$$\ddot{\mathbf{x}}_h = \mathbf{J}_h\ddot{\mathbf{q}} = \boldsymbol{\Lambda}_h^{-1}\mathbf{F}_h, \tag{B.9}$$

where $\boldsymbol{\Lambda}_h^{-1}$ is the inverse of the VT operational space inertia at the selected operational point. Note in (B.9) that the space of feasible motions at the selected operational point can be identified with the range space of $\boldsymbol{\Lambda}_h^{-1}$. Accordingly, the space of user motions hindered by the virtual joints can be identified with the null space of $\boldsymbol{\Lambda}_h^{-1}$.
From (B.8) and (B.9) it follows that:

$$\boldsymbol{\Lambda}^{-1}\mathbf{F}_h = \mathbf{J}_h\mathbf{D}^{-1}\left(\mathbf{J}_h^T\mathbf{F}_h + \mathbf{W}^T\boldsymbol{\lambda}\right), \tag{B.10}$$

where $\boldsymbol{\Lambda}_h^{-1}$ and $\boldsymbol{\lambda}$ are unknown. Therefore, the constraint equation (B.2) is used to compute the Lagrange multipliers (after substituting $\ddot{\mathbf{q}}$ from (B.1) and assuming negligible velocity terms):

$$\boldsymbol{\lambda} = -\left(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T\right)^{-1}\mathbf{W}\mathbf{D}^{-1}\mathbf{J}_h^T\mathbf{F}_h. \tag{B.11}$$

Substitution of (B.11) into (B.10) gives:

$$\boldsymbol{\Lambda}_h^{-1}\mathbf{F}_h = \mathbf{J}_h\mathbf{D}^{-1}\left(\mathbf{J}_h^T - \mathbf{W}^T\left(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T\right)^{-1}\mathbf{W}\mathbf{D}^{-1}\mathbf{J}_h^T\right)\mathbf{F}_h. \tag{B.12}$$

which must hold for all hand wrenches $\mathbf{F}_h$. Hence:

$$\begin{aligned}\boldsymbol{\Lambda}_h^{-1} &= \mathbf{J}_h\mathbf{D}^{-1}\left(\mathbf{J}_h^T - \mathbf{W}^T\left(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T\right)^{-1}\mathbf{W}\mathbf{D}^{-1}\mathbf{J}_h^T\right)\\ &= \mathbf{J}_h\mathbf{D}^{-1}\mathbf{J}_h^T\\ &\quad -\mathbf{J}_h\mathbf{D}^{-1}\mathbf{W}^T\left(\mathbf{W}\mathbf{D}^{-1}\mathbf{W}^T\right)^{-1}\mathbf{W}\mathbf{D}^{-1}\mathbf{J}_h^T.\end{aligned} \tag{B.13}$$

In (B.13), the first term is the inverse of the operational space inertia of the VT with the loops opened and the second term embeds the loop closure effects. Note the coupling between loop closure geometry and configuration space inertia for VTs with dynamics in extended generalized coordinates. Due to this coupling, the loop closure constraints are computed in this work alongside the other joint constraints via singular value decomposition (SVD) of $\boldsymbol{\Lambda}_h^{-1}$:

$$\boldsymbol{\Lambda}^{-1} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T. \tag{B.14}$$

Specifically, the directions of cut and regular joint constraints form the rows of $\mathbf{U}$.

B.5 Implementation

Because SVD algorithms do not have guaranteed running time, the computation of the singular directions of $\mathbf{\Lambda}^{-1}$ is sidestepped in the proposed implementation via decoupling the force control loop from the simulation through a local model of interaction [6] (see Fig. B.3). The local model is a reduced simulation that runs at the frequency of the force control loop. It approximates the interaction between the VT and the virtual environment through the interaction between the VT and nearby objects. The quality of the approximation is maintained via updating the local at each step of the virtual environment simulation.

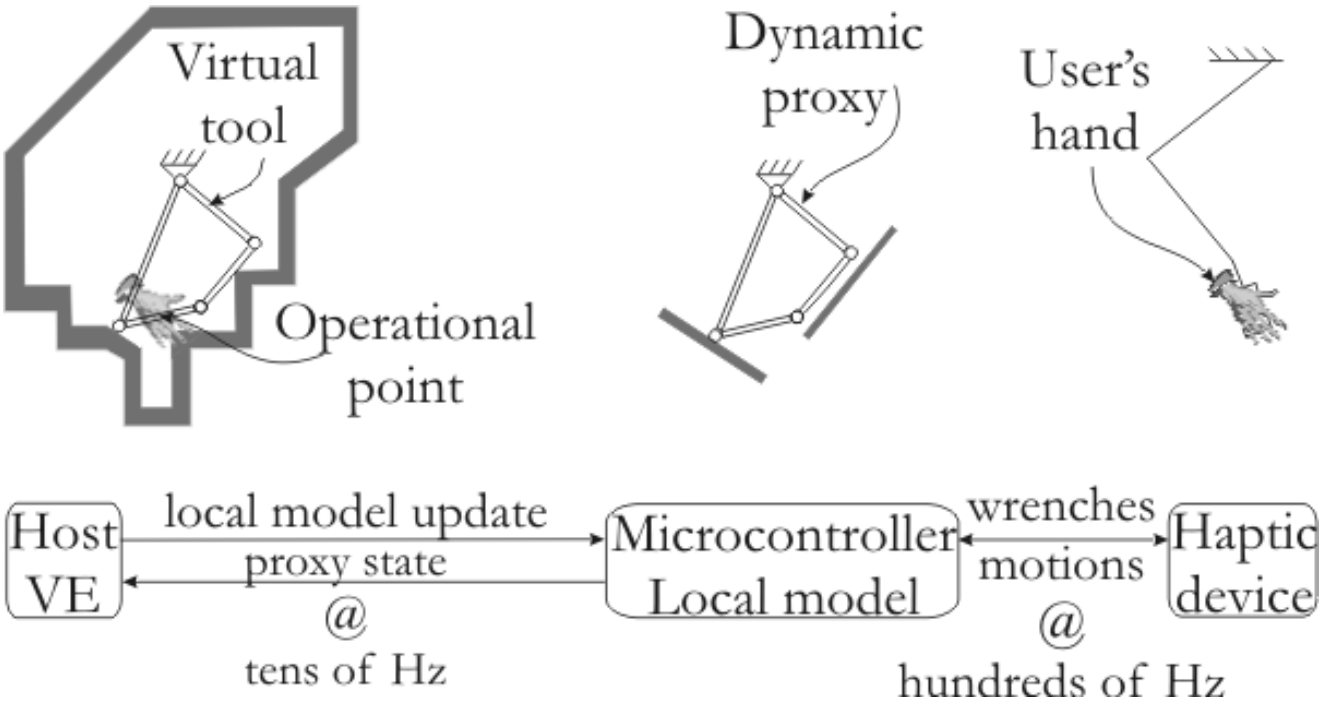


Figure B.3: Decoupling of the force control loop from the virtual environment simulation through a local model of interaction.

The decoupling permits computations to be distributed between the simulation/graphics and the haptics processors. Specifically, contact geometry, $\mathbf{\Lambda}^{-1}$ and the directions of joint constraint are computed on the graphics processor. The VT dynamics are computed on the haptics processor and are rendered to users as described in Section B.3. Further computational savings are achieved on the haptics processor by approximating $\mathbf{D}$, $\mathbf{W}$, and $\mathbf{w}$ in (B.6) with their values in the virtual environment at the moment of the update. Lastly, it is assumed that users manipulate the VT slow enough that the Coriolis and centripetal effects can be ignored (similar to work in [21] and [23]) and the VT dynamics are simulated by:

$$\begin{bmatrix} \widehat{\mathbf{D}} & \widehat{\mathbf{W}}^T \\ \widehat{\mathbf{W}} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \ddot{\mathbf{q}} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^{c} \mathbf{J}_i^T \mathbf{F}_i + \mathbf{J}_h^T \mathbf{F}_h - \widehat{\mathbf{G}} \\ -\widehat{\mathbf{w}}_{stab} \end{pmatrix} \tag{B.15}$$

In (B.15), $\widehat{\mathbf{D}}$, $\widehat{\mathbf{W}}$, $\widehat{\mathbf{w}}_{stab}$ and $\widehat{\mathbf{G}}$ denote the values of $\mathbf{D}$, $\mathbf{W}$, $\mathbf{w}_{stab}$ and $\mathbf{G}$, respectively, computed by the graphics engine and sent to the local model at the update. Furthermore, the directions of joint constraint are approximated through their values in the virtual environment at the update, by computing the SVD of $\mathbf{\Lambda}^{-1}$ on the graphics processor:

$$\widehat{\mathbf{\Lambda}}^{-1} = \widehat{\mathbf{U}}\widehat{\mathbf{\Sigma}}\widehat{\mathbf{V}}^T \tag{B.16}$$

and using the columns of $\widehat{\mathbf{U}}$ in the local model to estimate these directions. Equations (B.15) and (B.16) allow the VT dynamics to be simulated at the frequency of the haptic controller. Operation of VTs with closed loops using these approximations is demonstrated experimentally in the following section.

B.6 Experiments

This section illustrates manipulations of the VT depicted in Fig. B.4, whose parameters are given in Table B.1. A controlled constant wrench $\mathbf{F}_h = \begin{pmatrix} -0.5\text{N} & 0\text{N} & -0.0025\text{Nm} \end{pmatrix}^T$ [1] is applied at the centre of mass of the second link in the experiment (note link numbering in Fig. B.4). This constant wrench ensures the "same" user during successive manipulations. The VT is initially at rest in the position $\mathbf{q}_0 = \begin{pmatrix} \frac{\pi}{4}\text{rad} & -\frac{\pi}{2}\text{rad} & -\frac{4\pi}{10}\text{rad} & \frac{\pi}{2}\text{rad} \end{pmatrix}^T$, as depicted in Fig. B.4. The stiffness and damping used for rendering joint constraints are $k_j = 200\text{N/m/kg}$ and $b_j = 50\text{N/(m/s)/kg}$. Kinematic correspondence between the device and the VT are maintained via $\mathbf{K}_{pc} = \begin{pmatrix} 100\text{N/m} & 100\text{N/m} & 0.5\text{Nm/rad} \end{pmatrix}^T$ and $\mathbf{B}_{pc} = \begin{pmatrix} 70\text{N/(m/s)} & 70\text{N/(m/s)} & 0.375\text{Nm/(rad/s)} \end{pmatrix}^T$ [2]. The gains of the graphics controller are $\mathbf{K}_g = \begin{pmatrix} 1000\text{N/m} & 1000\text{N/m} & 100000\text{Nm/rad} \end{pmatrix}^T$ and $\mathbf{B}_g = \begin{pmatrix} 100\text{N/(m/s)} & 100\text{N/(m/s)} & 10000\text{Nm/(rad/s)} \end{pmatrix}^T$. Because the testbed virtual environment includes only the closed loop VT, the VT dynamics are approximated by:

$$\widehat{\mathbf{D}}\ddot{\mathbf{q}} = \mathbf{J}_h^T \mathbf{F}_h - \widehat{\mathbf{W}}^T \boldsymbol{\lambda} \tag{B.17}$$

in the local model of interaction.

Table B.1: VT parameters in the experimental manipulations.

Link length	**Link mass**	**Linak inertia**
$l_1 = 0.045$ (m)	$m_1 = 1$ (kg)	$I_1 = 0.0021$ (kg·m^2)
$l_2 = 0.045$ (m)	$m_2 = 1$ (kg)	$I_2 = 0.0021$ (kg·m^2)
$l_3 = 0.021$ (m)	$m_3 = 1$ (kg)	$I_3 = 0.0021$ (kg·m^2)
$l_4 = 0.060$ (m)	$m_1 = 0.5$ (kg)	$I_1 = 0.00105$ (kg·m^2)

The proposed method for computing $\boldsymbol{\Lambda}^{-1}$ and the directions of loop closure constraint is validated via two successive manipulations. During the first manipulation, the loop closure constraint is imposed through stiffness control by penalizing motion along the directions resisted by the cut joint. During the second manipulation, the loop closure costraint is rendered through impedance control by controlling the acceleration of the end-effector of the haptic device to zero along the directions of cut joint constraint. The experimental results are plotted in Figs. B.5 and B.6.

Figs. B.5(a) and B.6(a) depict the number of joint constraints imposed on the haptic interface by the VT and thus, identify the time during which the loop closure constraint is active in each of the two manipulations. These figures illustrate that, in the virtual environment, the virtual joints impose at least one constraint on the operational point throughout both manipulations. This is because the selected operational point is on a link with insufficient degrees of freedom. Hence, during either manipulation, the loop closure constraint is active ($\lambda \neq 0$) only when two joint constraints are active in the haptic simulation.

Figs. B.5(b) and B.6(b) plot the trajectories of the end-effector of the haptic device (HD) and of the selected operational point on the VT (OP). Although the different controllers change the VT dynamics and thus the OP trajectories during the two manipulations, a qualitative comparison of the results is possible. Note that the trajectory of the end-effector of the haptic interface follows the trajectory of the

[1] For the impedance type haptic interface used in the experiment, a constant wrench is a worst case approximation of the user for stability.

[2] These gains are much weaker than the contact stiffness and damping, implemented in the haptic simulation via $k_{contact} = 15,000\text{Nm}$ and $b_{contact} = 300\text{N/(m/s)}$.

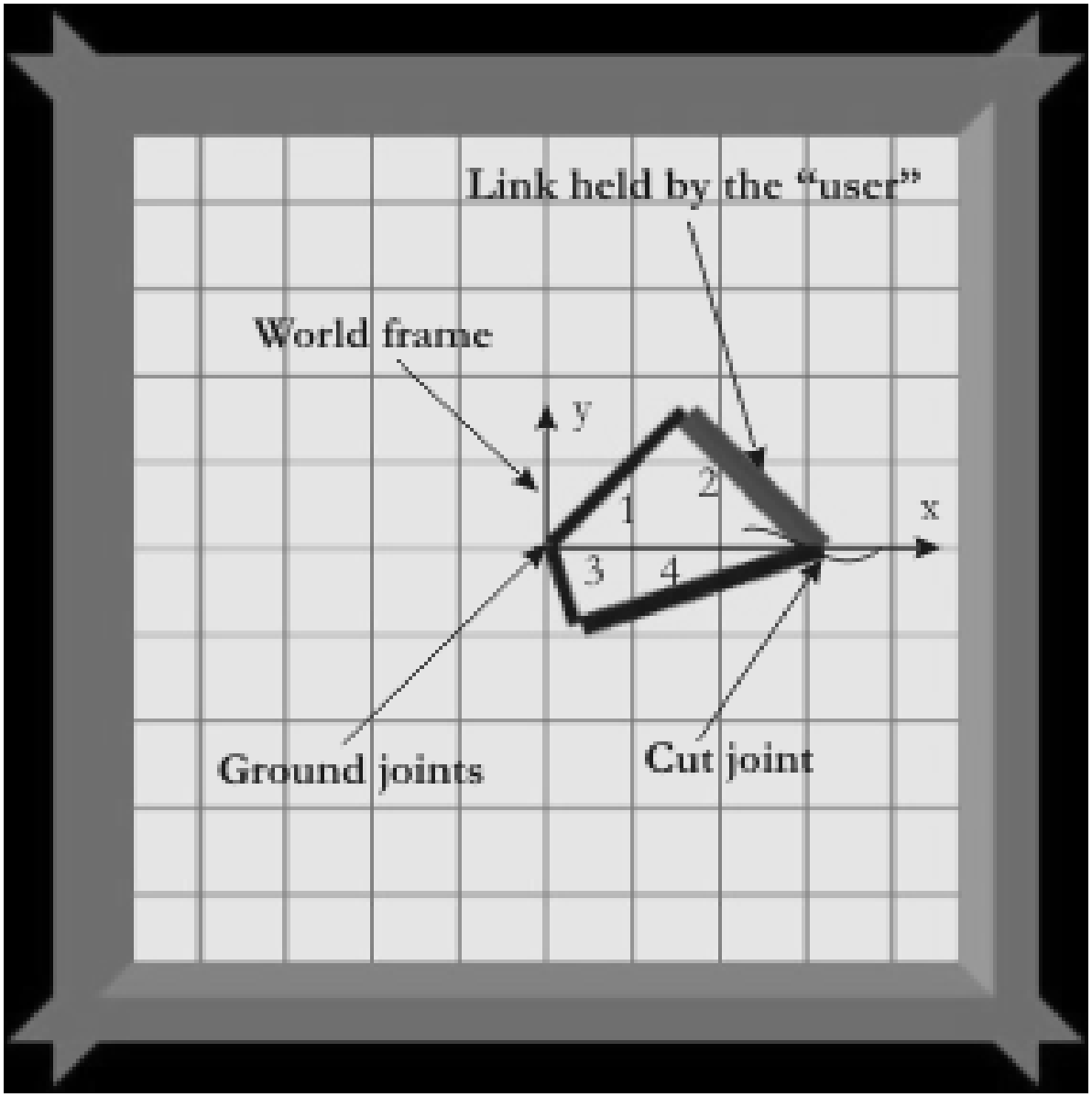
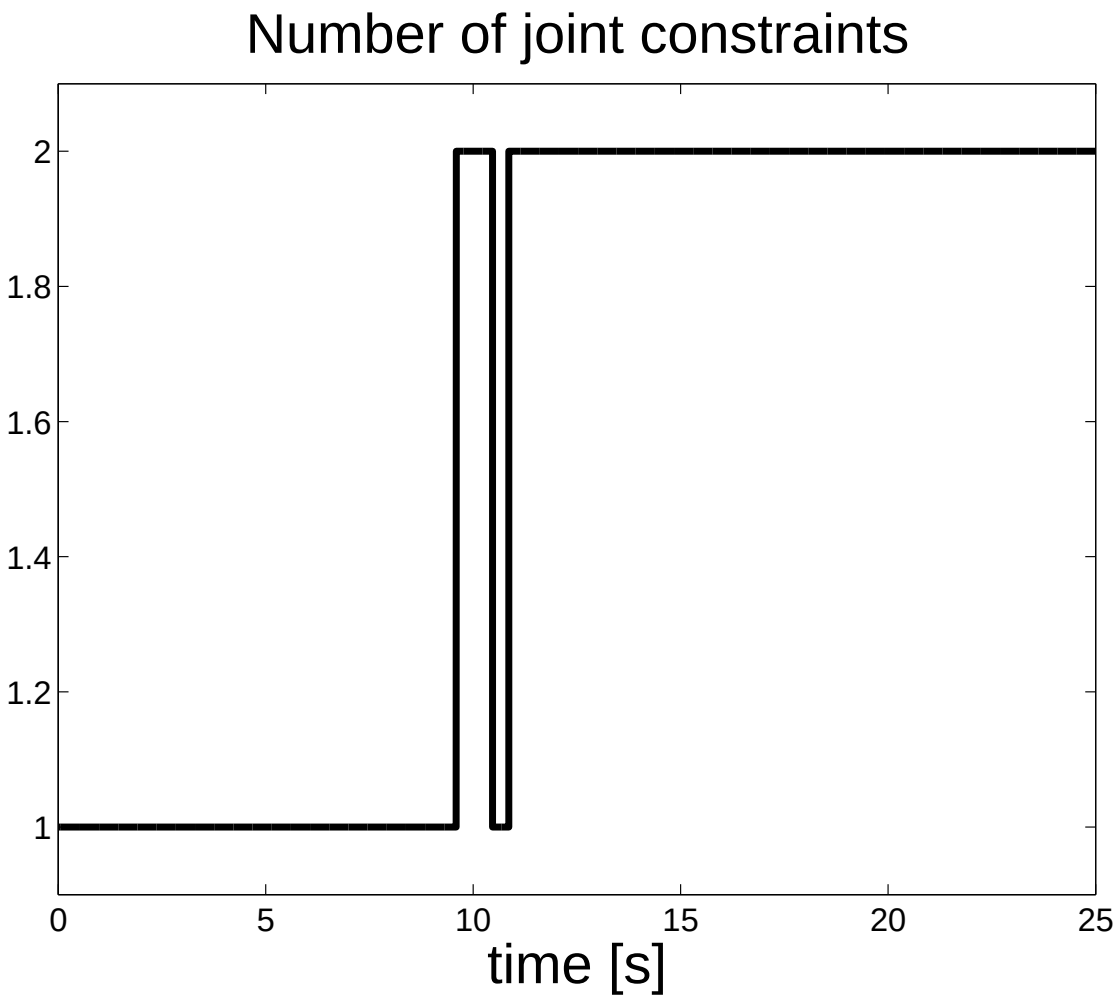
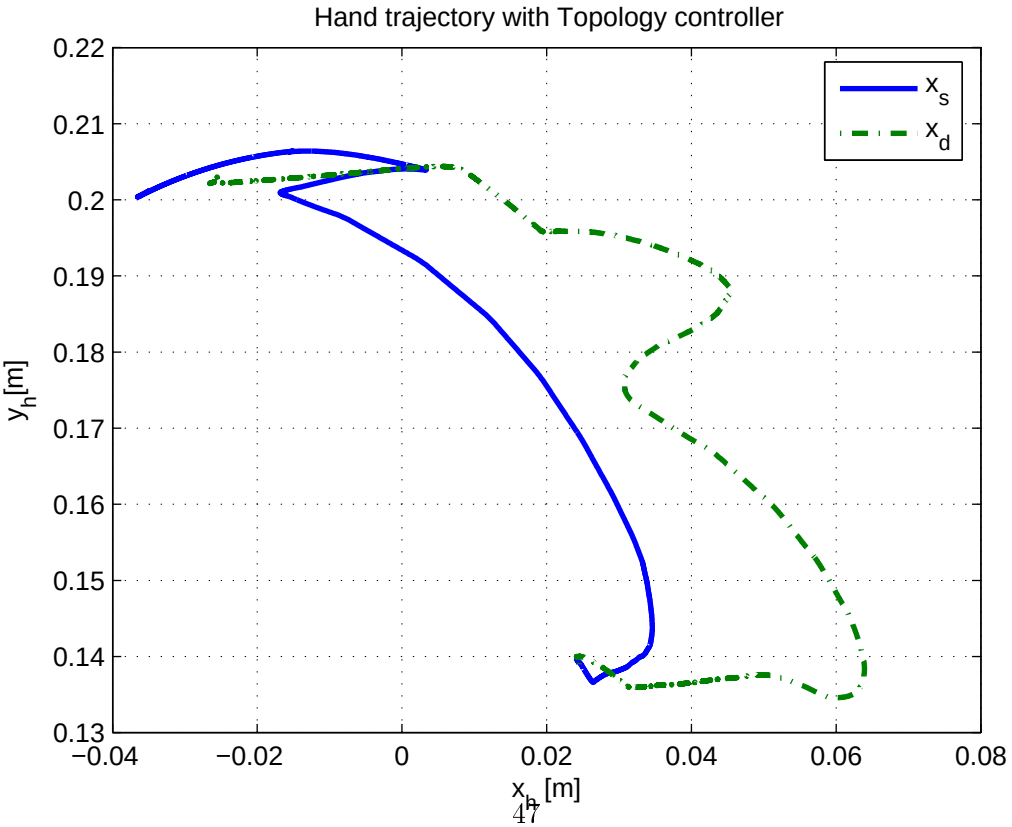


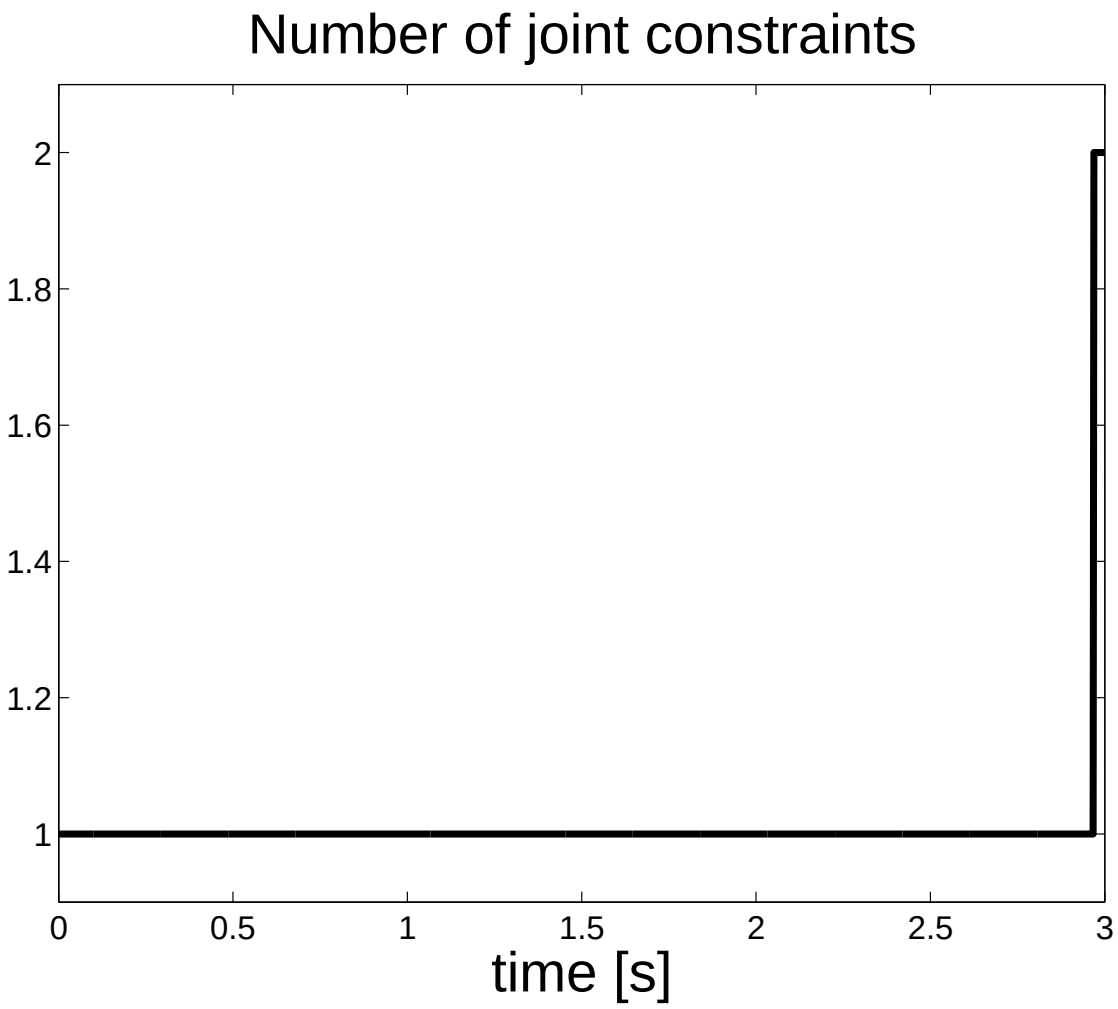
Figure B.4: Planar testbed virtual environment used to illustrate haptic manipulation of VTs with closed loops. Note link numbering and the cut joint.



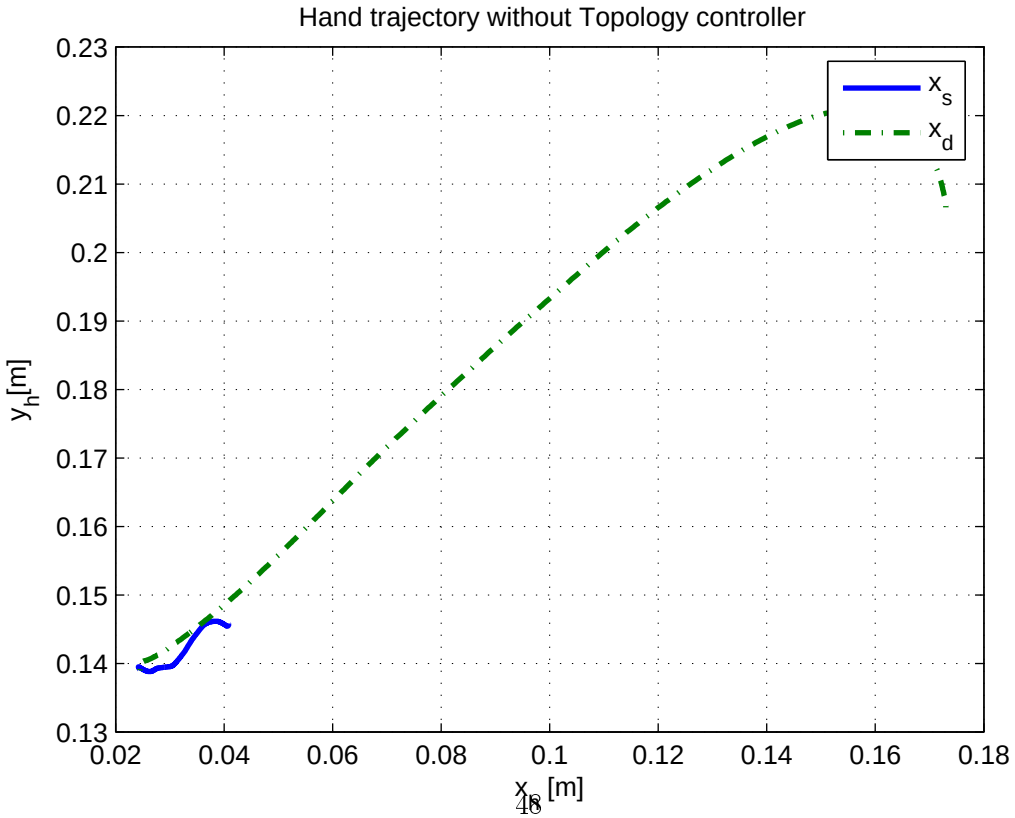
(a) Number of joint constraints imposed by the VT on the operational point.



(b) Experimental trajectories: of the user's hand on the haptic device (HD); and of the operational point



(a) Number of joint constraints imposed by the VT on the operational point.



(b) Experimental trajectories: of the user's hand on the haptic device (HD); and of the operational point

simulated operational point when joint constraints are rendered via stiffness control (Fig. B.5(b)). In contrast, the drift between the two trajectories is significant when joint constraints are imposed via impedance control (Fig. B.6(b))[3]. These results illustrate that stiffness control enforces the cut joint as effectively as the serial-chain joint. Hence, they validate the method proposed for computing $\mathbf{\Lambda}^{-1}$ and the loop closure constraint directions. The effect of the loop closure constraint on the device (i.e., on the user) motion is also clearly identifiable in Fig. B.5(c). Note in this figure that the wrench applying the joint constraints to the device (hand) is larger when the loop closure constraint is active. This larger wrench demonstrates that the stiffness of the two constraints combines to tighter confine the device to the simulated operational point.

B.7 Conclusion

This paper has proposed an approach to enabling users to manipulate virtual linkages with closed loops. In this approach, linkage dynamics are simulated in configuration space and loop closure constraints are maintained via Lagrange multipliers augmented with Baumgarte stabilization terms. The virtual dynamics are rendered to users in Cartesian space. The varying linkage inertia is applied via impedance control in operational space, and the joint contraints are enforced via stiffness control along directions orthogonal to operational space. The ability to restrict user's motion as required by the virtual closed loops has been validated through an experiment performed using a planar haptic interaction system. Future work will investigate: the haptic manipulation of active VTs, i.e., VTs under feedback control; the operation of VTs with redundant loop closure constraints; and the simultaneous manipulation of VT with closed loops by multiple users.

[3]The drift in Fig. B.6(b) is limited by ending the experiment before the device end-effector reaches the mechanical boundary of its workspace, after approximately 3s and just after the loop closure constraint becomes active.

Bibliography

[1] J. Baumgarte. Stabilization of Constraints and Integrals of Motion in Dynamical Syatems. *Computer Methods Appl. Mech. Eng.*, 1:1–16, 1972.

[2] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *The Numerical Solution of Intial Value Problems in Differentail-Algebraic Equations.* Elsevier, New York, NY, 1989.

[3] J.E. Colgate, P. E. Grafing, M. C. Stanley, and G. G. Schenkel. Implementation of Stiff Virtual Walls in Force-Reflecting Interfaces. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pages 202–208, 1993.

[4] D. Constantinescu, S. E. Salcudean, and E. A. Croft. Haptic Manipulation of Serial-Chain Virtual Mechanisms. *ASEM Transactions, Journal of Dynamic Systems, Measurement, and Control*, 128:65–74, 2006.

[5] D. Constantinescu, S.E. Salcudean, and E.A. Croft. Haptic Rendering of Rigid Contacts using Impulsive and Penalty Forces. *IEEE Trans. Robot.*, 21(3):309–323, 2005.

[6] D. Constantinescu, S.E. Salcudean, and E.A. Croft. Local Models of Interaction for Realistic manipulation of Rigid Virtual Worlds. *Int. J. Robot. Res.*, 24(10):789–804, 2005.

[7] D. Constantinescu, S.E. Salcudean, and E.A. Croft. Haptic Manipulation of Serial-Chain Virtual Mechanisms. *ASME Trans. J. Dyn. Syst. Meas. Contr.*, 128(1):65–74, 2006.

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

[9] R. Featherstone. Efficient Factorization of the Joint-Space Inertia Matri for Branched Kinematic Trees. *Int. J. Robot. Res.*, 24(9):487–500, 2005.

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

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

[12] R.B. Gillespie. On-Line Symbolic Constraint Embedding for Simulation of Hybrid Dynamical Systems. *Multibody Dyn.*, 14:387–417, 2005.

[13] P. Gohari. Lecture 6: Discrete equivalents. *Course: ELEC 6061*, 2006.

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

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

[16] P. Mitra and G. Niemeyer. Dynamic Proxy Objects in Haptic Simulations. In *Proc. IEEE Conf. Robot. Autom. Mechatronics*, pages 1054–1059, 2004.

[17] A. Nahvi, D.D. Nelson, J.M. Hollerbach, and D.E. Johnson. Haptic Manipulation of Virtual Mechanisms from Mechanical CAD Designs. In *Proc. IEEE Int. Conf. Robot. Autom.*, pages 375–380, Leueven, Belgium, May 1998.

[18] S. Nicosia and P. Tomei. Robot control by using only joint position measurements. *IEEE Transactions on Automatic Control*, 35(9):1058–1061, 1990.

[19] P. E. Nikravesh. *Computer-Aided Analysis of Mechanical Systems.* Prentice-Hall, Englewood Cliffs, NJ, 1988.

[20] P. E. Nikravesh. Systematic Reduction of Multibody Equations of Motion to a Minimal Set. *Int. J. Non-Linear Mech.*, 25(2/3):143–151, 1990.

[21] D. Ruspini and O. Khatib. Dynamic Models for Haptic Rendering Systems. In *Adv. Robot Kin.: ARK98*, pages 523–532, Strobl/Salzburg, Austria, 1998.

[22] D. Ruspini and O. Khatib. Collision/Contact Models for Dynamic Simulation and Haptic Interaction. In *Proc. Int. Symp. Robot. Res.*, pages 185–195, Snowbird, UT, 1999.

[23] D. Ruspini and O. Khatib. Haptic Display for Human Interaction with Virtual Dynamic Environments. *J. Robot. Syst.*, 18(2):769–783, 2001.

[24] K. Salisbury and F. Conti. Haptic Rendering: Introductory Concepts. *IEEE Computer Society*, pages 24–32, 2004.

[25] DiMaio S.P. Salcudean S.E. Abolmaesumi P. Sirouspour, M.R. and C. Jones. Haptic Interface Control- Design Issues and Experiments with a Planar Device. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 789–794, San Frnacisco, CA, 2000.

[26] N. van de Wouw. *Multibody Dynamics, 4J400.* TU/e, 2004.