23.1 Introduction

A haptic interface is a motorized and instrumented device that allows a human user to touch and manipulate objects within a virtual environment. As shown in Figure 23.1, the haptic interface intervenes between the user and virtual environment, making a mechanical contact with the user and an electrical connection with the virtual environment. At the mechanical contact, force and motion are either measured by sensors or driven by motors. At the electrical connection, signals are transmitted that represent the force and motion occurring at a simulated mechanical contact with a virtual object. A controller within the haptic interface processes the various signals and attempts to ensure that the force and motion signals describing the mechanical contact track or in some sense follow the force and motion signals at the simulated contact. The motors on the haptic interface device provide the authority by which the controller ensures tracking. A well-designed haptic device and controller will cause the behaviors at the mechanical and simulated contacts to be "close" to one another. In so doing it will extend the mechanical sensing and manipulation capabilities of a human user into the virtual environment.

23.1.1 Related Technologies

Naturally, the field of haptic interface owes much to the significantly older field of telerobotics. A telerobot intervenes between a human user and a remote physical environment rather than between a user and a computationally mediated virtual environment. In addition to a master manipulator (to which the haptic
interface device is analogous), a telerobot includes a slave manipulator that makes mechanical contact with the remote environment. Telerobots were first developed in the 1940s for radioactive materials handling [1] and in their early realizations, were purely mechanical linkages designed to extend the touch and manipulation capabilities of a human operator past a protective wall. When electromechanical control first made its way into telerobots, only the slave was motorized so that it could follow the motions picked up by sensors in the master. But without a motorized master, the touch and manipulation capabilities of an operator were diminished when compared with the mechanical linkage, because interaction forces at the slave could not be felt. Later, with the introduction of motors on the master, forces (or motion errors) picked up by sensors on the slave could be reflected back to the operator. Thus the notion of a bilateral or force-reflecting teleoperator came into being, and the full manipulation and feel capacities of the mechanical linkage were restored. In analogy to the development of the bilateral telerobot, virtual reality gloves and other pointing devices support the positioning of virtual objects, but not until such devices are also outfitted with motors can the user feel interaction forces reflected back from the virtual environment. Once outfitted with motors, these devices graduate from mere pointing tools to become haptic interfaces.

Another prior technology related to the field of haptic interface is flight simulation. Flight simulators intervene between a human pilot trainee and a virtual aircraft and airfield using a yoke or joystick input device, a visual display, and often a motion display. See [2] for a full review of this technology. Haptic interface and flight simulation share much in terms of the virtual environment and real-time simulation technology, but an important difference exists with regard to the kind of interface made up by an input and display device. A haptic interface is simultaneously an input and display device, with two-way information flow across a single mechanical contact. In contrast, the motion display that moves the seat or cockpit and the visual display in a flight simulator do not share the same port with the yoke or joystick input device. If the yoke or joystick is motorized (or hydraulically driven) so that the pilot-trainee can feel the loads on the simulated aircraft control surfaces, then that yoke or joystick is actually a haptic interface. Indeed, such technology, both in flight simulation and fly-by-wire aircraft [3], pre-dates the field of haptic interface by many years.

### 23.1.2 Some Classifications

The term *haptic* derives from the Greek word *haptein*, meaning “to touch”1 and is used broadly in engineering to describe mechanical interaction, especially between a human and machine. Naturally,

---

1The verb *haptein* is an archaic version of the modern Greek verb *ἀπατεῖσθαι* (“to touch”) that in turn is derived from the modern noun *ἀφήν* (“touch” or “feeling”). The phonetic spellings of the verb *haptein* and noun *haptein* propagated in English circa 1890, with an “h” in front of the initial Greek “α” to denote the accent grave.
mechanical interaction encompasses a very rich set of behaviors, including actions that may be planned and perceived quantities that may be remembered.

As depicted in Figure 23.2, mechanical interactions may be roughly categorized by the percepts involved, whether tactile (served by sensors located in the skin, called cutaneous mechanoreceptors) or kinesthetic (served by sensors in the muscles and joints: muscle spindles and Golgi organs). However, haptic interaction also includes manipulation, a process that involves both perception and action. Therefore, a strict classification in terms of tactile and kinesthetic sensing is difficult to achieve. Nevertheless, haptic interface devices may be roughly categorized as tactile or kinesthetic interfaces.

Tactile devices generally involve multiple, individually motor-driven localized contacts that describe an area contact with the skin, invoking texture and shape percepts and supporting haptic explorations such as stroking and contour-following. Kinesthetic interfaces, though they may also involve area contact with the skin, use a single motor-driven rigid body such as a knob, stylus, or thimble that is touched or grasped in the user's hand. The paradigm for interaction with virtual objects is then through that rigid body, an image of which is projected into the virtual environment. Thus, kinesthetic interfaces are used to invoke relationships between force and motion that pertain to interactions between a single rigid body and the virtual environment, with percepts such as stiffness, damping, and inertia. Kinesthetic interfaces support haptic explorations such as pushing, squeezing, indenting, and throwing. Kinesthetic interfaces may be further classified as to whether they are grounded (include a mechanical linkage to a fixed table or floor) or ungrounded (lacking such a linkage). Ungrounded interfaces are capable of producing only inertia forces and torques, whose frequency response lacks a DC component. For example, an ungrounded motor with an eccentric on its shaft such as one might find in a cell phone vibration ringer can be used to produce certain haptic or so-called vibrotactile impressions. In the present chapter, we will concentrate on kinesthetic haptic interfaces. For a full overview of haptic device technology, covering both tactile and kinesthetic interfaces, see [4].

23.1.3 Applications

Applications of haptic interface include training and assessment of cognitive and motor skills, rehabilitation, scientific visualization, and entertainment. Industries in which haptic devices are beginning to appear include medical training (e.g., surgical skill training simulators), automotive (so-called infotronic devices or haptic interfaces to climate and other controls that keep the eyes on the road), and gaming industries (immersive and shared virtual environments). Entertainment is one of the fastest growing markets for commercial development of haptic interface, where the ability to engage more of the user’s senses leads to new dimensions of interactivity. A number of industrial concerns are quickly growing to serve these markets. Also, because human haptic psychophysics is a relatively new and very rich field of academic and industrial research, a small but significant commercial activity has grown to provide devices for basic and applied haptics research.

Additional uses of haptic interface are on the horizon. For example, haptic interface can be used in an augmented reality approach to overlay mechanical features or virtual fixtures on one's interaction with physical objects. The user can use these virtual fixtures to develop more efficient, more ergonomic, or
more robust strategies for manipulating objects. For example, a virtual fixture might be used to eliminate tremor when performing retinal surgery through a teleoperator [5].

Closely related to the idea of virtual fixtures is the use of haptic interface to realize shared control between a human and an automatic controller. A human and an automatic control system can collaborate to complete a task if they both apply their control inputs as forces on a single manual interface (e.g., joystick or steering wheel). The user can monitor the actions of the automatic controller by feel using his hand resting on the joystick or he can augment or override the actions of the automatic controller by applying his own forces. Shared control through a haptic interface is being explored in the context of driver assist for agricultural and road vehicles [6] and surgical assist [5, 7].

Thinking more generally about the training of motor skills, a haptic interface can present the force-motion relationship (including history-dependent or dynamical relationship) that is deemed appropriate or representative of the physical world, yet carries no risk when errors are committed. Alternatively, a force-motion relationship that is adapted to the user, or contains specially designed responses to errors, may be created. The particulars of the mechanical interaction are all programmable, as is the relationship to the accompanying visual and audio responses. Further, a haptic interface can be used to test various hypotheses in human motor control, the answers to which may be used to develop therapies for rehabilitation or motor skill training [8].

Haptic interface has also been used to teach cognitive skills or concepts in certain disciplines. For example, concepts in system dynamics for engineering undergraduates can be demonstrated in a hands-on manner with the hopes that physical intuition might kick-in to boost understanding and the development of good modelling skills [9, 10]. By programming their own virtual environments using the differential equations studied in class and then interacting with them through a haptic interface, students should be able to more quickly establish appropriate relationships between mathematical models and observed mechanical behavior.

### 23.2 An Overview: System Performance Metrics and Specifications

A haptic interface, precisely because its role is to intervene between a user and virtual environment, is difficult to evaluate by itself. Its performance in a given application depends on certain characteristics and behaviors of the human user (e.g., size of hand, grip, experience) and on certain properties of the virtual environment (e.g., update rate, model fidelity, integration with visual display). What really determines the success of a given design is, of course, the performance of the coupled system comprising all three elements: user, haptic interface, and virtual environment. Yet, performance measures that are tied to the haptic interface alone are much to be preferred over system measures, for purposes of modularity. With measures defined only in terms of the haptic interface, one could design for any user and then interchange various haptic devices and virtual environment rendering algorithms. Also, such metrics are necessary to compare devices with one another. To define such haptic interface performance standards and to relate them to the system performance that really counts is one of the chief challenges of this field.

In the body of this chapter, performance measures to be attached to the haptic interface alone will be presented and discussed in detail. However, it is necessary to keep in mind that the most relevant performance measures in a given application depend on a user performing a particular task through a haptic interface on objects within a virtual environment. The contributions of the specification of the task, the selection of a particular user, and the use of a particular virtual environment rendering algorithm to the determination of the system performance cannot be underestimated. The haptic interface is only one of several determinants. Examples of system performance measures include: discriminability of two rendered textures or impedances, pursuit tracking performance, and peg-in-hole task completion times. Production of these performance measures requires some form of experiment with human subjects and a particular virtual environment.
To emphasize the roles of the user and virtual environment in system performance, the central section of this chapter (whose topic is haptic interface metrics) is flanked by a section on human haptics and a section on virtual environment design. Thus, the organization of this chapter reflects the three-element system comprising human, haptic interface, and virtual environment. But before launching into these three sections, a quick overview with additional unifying commentary is given here.

### 23.2.1 Characterizing the Human User

Truly it is the human user whose role is central and ties together the various applications of haptic interface. A very useful viewpoint is that the ground for the success of virtual reality (and by extension haptic interface) was laid by the failure of robotics. So far, the robot has not been able to enter the world of the human. Although the assembly line and paint shop have been opened to robots, by comparison the significantly less structured environments of the kitchen and laundry room have remained closed. Coping with the complexities of the physical world has proven quite difficult in practice. On the other hand, the human can easily suspend disbelief and adapt to synthetic worlds and, thus, in a very real sense, can enter the world of the robot. In most successful applications or demonstrations of haptic interface to date, the adaptability and forgiving nature of the human were harnessed, notwithstanding the various technical challenges and technical achievements that were also brought to bear. Thus, a key to design for any given application is an understanding of human haptic perception and its cooperative role with other perceptual modalities.

Likewise, the significant ability of a human when acting as a controller to adapt to new circumstances and respond to unforeseen events has given rise to the use of robots that extend human capabilities rather than replace them. Thus, the field of teleoperation is ever active despite advances in autonomous robots. Even when restricted through limited kinematics, masking dynamics, and destabilizing time-delays, human operators acting through a telerobot are preferred over autonomous robots for many applications. Haptic interface draws motivation for its existence from the capabilities (even needs) of human users to suspend their disbelief, adapt their manipulation strategies, and perform within an otherwise inaccessible or unsafe training environment.

In Section 23.3 below, a small review of human haptic psychophysics will be undertaken. Further results from this branch of cognitive psychology will be discussed, especially those that impact the engineering of haptic interface devices and virtual environments.

### 23.2.2 Specification and Design of the Haptic Interface

Figure 23.3 shows the three elements: human user, haptic interface, and virtual environment presented in both a network diagram and a block diagram. Associated with both diagrams are the user/interface

![Diagram of System Network and Block Diagram](image-url)
interaction force \( F_h \) and velocity \( v_h \) and interface/virtual-environment interaction force \( F_e \) and velocity \( v_e \). Note that in the network diagram, a sign convention is specified for each variable that is not explicitly available in the block diagram. In the block diagram, a causality or signal flow direction is assumed that is not explicitly available in the network diagram. The sign convention and causality assumption are both choices made by the analyst or modeler.

Certainly the sign convention is arbitrary and has no implication on the eventual system equations. The causality assumption, however, by specifying what variables are inputs and outputs for a particular component, determines the form, or what quantities are on the left hand sides of the system equations. Although causality is chosen by the analyst, the particular causality assumption shown in the block diagram of Figure 23.3 is most appropriate for a haptic device designed according to the impedance display paradigm and a virtual environment in a forward dynamics form. The opposite causality is available for both the haptic interface and virtual environment, with corresponding modifications to the hardware. If the haptic interface uses motors in current-control mode to apply forces and moments, it is generally considered to be an impedance display. A haptic interface that uses force or torque sensors to close a control loop around the interface/human interaction force is called an admittance display. Most haptic interface devices to date are controlled using impedance display, which may be implemented without force or torque sensors using low inertia motors and encoders connected to the mechanism through low friction, zero-backlash transmissions with near-unity mechanical advantages.

A natural way to analyze the haptic interface independent of the user and virtual environment is to look at quantities defined at the boundaries: the two mechanical contacts, the physical contact with the user and the simulated contact with the virtual environment. One can define and analyze the notion of “closeness” between the force-motion relationship observed at the physical contact and the force-motion relationship observed at the virtual contact. One useful viewpoint is that the haptic interface is a kind of filter on top of the virtual environment, transmitting certain aspects of mechanical behavior of the virtual environment, but also adding some of its own behavior to what the user feels.

Given linear models of virtual environment impedance, definitions of closeness are not difficult to pin down. For example, transparency has been defined for teleoperators as the ratio of the filtered mechanical impedance presented by the master to the impedance presented by the remote environment [11]. Transparency for a haptic interface can be defined as the ratio of the impedance presented by the device to that presented by the virtual environment. Thus transparency is the ability of the haptic interface to faithfully reproduce what is rendered by the virtual environment. If the user feels he is touching the virtual environment directly, then the force and motion measured at the physical contact should have the same relationship as that measured in the simulated contact with the virtual environment. After having defined “closeness” and determined some of the underlying factors, the objective is to optimize a given haptic device and controller design while satisfying certain implementation or cost limits.

When a user contacts a haptic interface, not only is one control loop closed as might be suggested in Figure 23.3, but multiple loops are closed. The human is after all a complex system containing its own hierarchically organized control system. One of the loops closed by the human around the haptic interface is quite tight, or of relatively high bandwidth. It does not involve any perception or voluntary action on the part of the user. That is the loop closed by the mechanics of the human body, taken without its supervisory sensory-motor controller. In other words, it is the loop closed by the passive dynamics of the user, or the nominal mechanical impedance of his hand or hand and arm. This loop determines certain aspects of the force-motion relationship at the user/interface contact, especially the high-frequency aspects. Oscillations that appear in the system with characteristic frequencies beyond the capabilities of human motor control, especially those that are otherwise absent when the user does not contact the interface, will depend only on this inner loop (the passive dynamics) and not the outer loop involving sensory-motor processes.

More will be said about such oscillations in Section 23.4 below. Here it suffices to say that these oscillations are associated with a stability limit and determine the limits of performance through a performance/stability tradeoff. An associated transparency/stability tradeoff has been established for teleoperators [11]. As in the field of teleoperation, the quantification of this tradeoff is one of the chief challenges in the field of haptic interface. Unlike transparency, which can be defined without specifying the human user and
virtual environment, stability is a system property. It is difficult to determine the limits of stability in terms independent of the dynamics of the human user and virtual environment. The work in this area will be reviewed in Section 23.4.

### 23.2.3 Design of the Virtual Environment

The virtual environment is an algorithm that produces commands for the haptic device in response to sensor signals so as to cause the haptic device to behave as if it had dynamics other than its own. As such, the virtual environment is a controller, while the haptic interface is the plant. Naturally, the algorithm is implemented in discrete time on a digital computer and together with the haptic interface and user, the interconnection forms a sampled-data system.

Figure 23.4 presents a schematic view of a haptic interface, a virtual environment, and the link between the two. In the top portion of the Figure, mechanical interaction takes place between the user's fingers and the haptic device end-effector. In the lower portion, an image of the device end-effector \( E \) is connected to an end-effector proxy \( P \) through what is called the *virtual coupler*. The proxy \( P \) in turn interacts with
objects such as A and B in the virtual environment. Proxy P might take on the shape of the fingertip or a tool in the user’s grasp.

The virtual coupler is depicted as a spring and damper in parallel, which is a model of its most common computational implementation, though generalizations to 3D involve additional linear and rotary spring-damper pairs not shown. The purpose of the virtual coupler is two-fold. First, it links a forward-dynamics model of the virtual environment with a haptic interface designed for impedance-display. Motion (displacement and velocity) of the virtual coupler determines, through the applicable spring and damper constants, the forces and moments to be applied to the forward dynamics model and the equal and opposite forces and moments to be displayed by the haptic interface. Note that the motion of P is determined by the forward dynamics solution, while the motion of E is specified by sensors on the haptic interface. The second role of the virtual coupler is to filter the dynamics of the virtual environment so as to guarantee stability when display takes place through a particular haptic device. The parameters of the virtual coupler can be set to guarantee stability when parameters of the haptic device hardware are known and certain input-output properties of the virtual environment are met. Thus, the virtual coupler is most appropriately considered part of the haptic interface rather than part of the virtual environment [12, 13]. If an admittance-display architecture is used, an alternate interpretation of the virtual coupler exists, though it plays the same two basic roles. For further discussion of the role played by the virtual coupler in performance/stability tradeoffs in either the impedance or admittance-display cases, see [12–14].

Another note can be made with reference to Figure 23.4: rigid bodies in the virtual environment, including P, have both configuration and shape — they interact with one another according to their dynamic and geometric models. Configuration (including orientation and position) is indicated in Figure 23.4 using reference frames (three mutually orthogonal unit vectors) and reference points fixed in each rigid body, while shape is indicated by a surface patch. Note that the end-effector image E has configuration but no shape. Its interaction with P takes place through the virtual coupler and requires only the configuration of E and P.

Figure 23.5 shows the user, haptic device, virtual coupler, and virtual environment interconnected in a block diagram much like the block diagram in Figure 23.3. Intervening between the human and haptic device, that “live” in the continuous, physical world, and the virtual coupler and virtual environment, that live in the discrete, computed world, are a sampling operator T and zero-order hold. The virtual coupler is shown as a two-port that operates on velocities \( v_m \) and \( v_e \) to produce the motor command force \( F_m \) and force \( F_e \) imposed on the virtual environment. Forces \( F_m \) and \( F_e \) are usually equal and opposite. Finally,
the virtual environment is shown in its forward dynamics form, operating on applied forces \( F_e \) to produce response motion \( v_e \). Naturally, the haptic device may use motors on its joints, so the task-space command forces \( F_m \) must first be mapped through the manipulator Jacobian before being applied to the motors.

Not apparent in the block diagram is the detail inside the virtual environment block which must be brought to bear to simulate systems with changing contact conditions, including a forward dynamics solver, collision detector, and interaction response algorithm. A collision detector and interaction response algorithm are needed to treat bodies that may collide, rest, slide, or roll on one another.

The virtual environment can easily include memory or the storage of information about past inputs, perhaps in the form of state variables. Thus, a virtual environment can emulate the behavior of a dynamical system or a physical system with energy storage elements. One very convenient way to produce such a virtual environment is to use numerical integration of the differential equations of motion of the modelled dynamical system. Since the integration must run in real-time, explicit numerical routines must be used.

As in the choice of any modelling formalism, numerous choices must be made in the design of a virtual environment, many of which have implications on the set of objects or kinds of interactions that may be simulated. The consequences of each choice are often quite complex and sometimes surprising. Unexpected or nonphysical behaviors sometimes appear, due to inherent limits of a formalism or certain assumptions that are challenged by the range of interaction behaviors that arise under human exploration. Tight computational resources imposed by the real-time requirement mean that the production of efficient and extensible algorithms is a very active field of research. Some of the problems and methods of this field will be explored in Section 23.5 below.

This concludes an overview. Another three sections appear below with the same headings: human haptics, haptic interface, and virtual environment. But this time each appearance treats its topic in a bit more technical detail and in a manner more independent of the others.

### 23.3 Human Haptics

Haptics research is concerned with understanding the way in which humans gather information from their environment through mechanical interaction. The field of human haptics lies primarily in the domain of cognitive psychology and has many proponents, including Lederman and Klatzky [15–17]. Contributions to the field of haptics follow and are often parallel to earlier contributions in the areas of vision and audition. Understanding haptic perception, however, is significantly more complicated (and interesting) than visual or aural perception, because the haptic sensory apparatus is integrated into the same organ (the hand, or more generally, the body) as the motor apparatus that is used to generate the mechanical stimuli. Haptic perception cannot be divorced from the process of manipulation. Mechanical excitation of objects in the environment is generally accomplished through muscle action so as to generate the sensory stimuli that carry the information. Katz gave this process of coupled motor control and haptic perception the name *active touch* in 1925 [18]. Gibson further developed the idea of active touch in his treatise on “ecological perception” in 1966 [19].

#### 23.3.1 Some Observations

An important theme throughout the development of virtual environments has been the resolution of the cost/benefit tradeoff of including another display technology. This is an especially important question when it comes to investing in haptic display technology, which is typically expensive and difficult to maintain. This motivates a thorough investigation into the value of haptic feedback. How, exactly, is haptic feedback used by a human user to form images in his mind, or to confirm and make more compelling the images first inspired by the other senses? And further, how is haptic feedback used by a human user to create and maintain manipulation strategies as he attempts to carry out an action on an object or set of objects in his environment?

Although very simple, some experiments by Schaal and Sternad [20, 21] shed light on the value of haptic feedback while juggling a ball. In these experiments, human subjects attempted to maintain a stable juggle
of a ball on a paddle in their control while being provided selectively with visual feedback, haptic feedback, or both. It was shown that a stable juggle was best maintained when haptic feedback was available, and the case of haptic feedback alone was much better than visual feedback alone and almost as good as combined visual and haptic feedback. It seems to be not just the continuous interaction dynamics, but also the intermittent impulses and changes in dynamics which inspire the most compelling images and inform as to the best strategies for manipulation.

Recent research in haptics has led to an understanding of certain aspects of the relationship between haptic perception and motion production. For example, Klatzky and Lederman [22, 23] showed that subjects will employ characteristic motor patterns when asked to ascertain certain object properties. Klatzky and Lederman called these motor patterns exploratory procedures. For example, when asked about an object's texture, subjects will glide or rub their fingers over the object, whereas when asked about shape, they will follow contours with fingertips or enclose the object between their hands. It seems that certain patterns of movement maximize the availability of certain information. Klatzky and Lederman have also conducted experiments on the recognition of object representations that have demonstrated poor apprehension of form in two dimensions but good apprehension in three dimensions [24].

Modern researchers in haptics are fortunate to have haptic interface devices available to them, much like researchers in audition have speakers and researchers in vision have computer displays. Previous researchers in the field, though restricted to experiments and observations with physical objects, nevertheless have produced much of our understanding. Perspective on the history of haptics research and thinking can lead to very valuable insights for haptics researchers, especially engineers working in the field. For this reason, a brief historical overview is given here.

23.3.2 Some History in Haptics

Haptics as an academic discipline dates to the time of Aristotle. His treatise De Anima [25], which dates from 350 B.C., still provides very interesting reading to a modern haptics researcher. To Aristotle, touch was the most essential of the five senses and is the one feature that can be used to distinguish an animal from a plant or an inanimate object. While some animals cannot see or hear, all respond to touch. Interestingly (in light of modern thought on haptics), there exists a thread tying together the sense of touch and the capacity for movement in Aristotle's work. To Aristotle, features are closely tied to function so that one may suitably classify an object by describing either its features or its function. Having identified the sense of touch as the distinguishing feature of animals, Aristotle associated touch with the accepted functional definition of animals: objects that move of their own volition. Today the close link between motility and haptics is readily acknowledged, both because the mechanical senses are indispensable in the production of movement and because movement is indispensable in the gathering of haptic information. Another question in haptics that interested Aristotle persists even today: Is skin the organ of touch or is the touch organ situated somewhere else, possibly deeper? Even Aristotle acknowledged that “We are unable clearly to detect in the case of touch what the single subject is which underlies the contrasted qualities and corresponds to sound in the case of hearing.”

In 1749, Diderot (of Diderot's Encyclopedia fame) published his “Letter on the Blind,” a fascinating account of tactile perception in the congenitally blind. He laid the foundation for our understanding of sensory substitution, that one sense gains in power with use or loss of another. Modern neurological evidence also points to the plasticity of the brain: changes in cortical organization occur with changes in use or type of sensory stimulation. Diderot also wrote on the role of memory and the process of learning in touch, noting that an impression of form relies on retention of component sensations.

Ernst H. Weber introduced systematic experimental procedures to the study of haptics and the other senses and is, thus, considered the founder of the field of psychophysics. His famous law, formulated while investigating cutaneous sensation, was reported in The Sense of Touch (1834). Weber's law states that one's ability to discriminate differences between a standard and a comparison is a function of the magnitude of the standard. For example, a larger difference is needed to discriminate between two weights when the standard weighs 100 g than when the standard weighs 20 g. Anticipating later work in haptics,
Weber recognized the role of intentional movement in the perception of hardness and distance between objects.

In 1925 David Katz published his influential book Der Aufbau der Tastwelt (The World of Touch) [26]. He was interested in bringing the sense of touch back into prominence, because psychological research in vision and audition had already outstripped haptics research. Although Katz was certainly influenced by the work of his contemporaries in Gestalt psychology, he was more concerned with texture and ground than form and figure. Rather than simplicity of the internal response, he was interested in the correspondence of the internal response with the external stimulus. But, consistent with Gestalt thinking, he held that sensations themselves are irrelevant. Rather, the invariants of the object are obtained over time, and an internal impression is formed that is quite isolated from the sensory input.

Katz was particularly interested in the role of movement in haptic perception. Resting your hand against a surface, you may feel that it is flat, but until there is relative movement between your fingertips and the surface, you will not be able to discern its texture. Only with movement do objects “come into view” to the haptic senses. With movement, touch becomes more effective than vision at discerning certain types of texture.

Katz noted that the pressure sense can be excluded by holding a stick or stylus between the teeth and moving it across some material: vibrations are still produced and accurate judgments can be made as to the material “touched.” Katz’s experiments with styli further suggest that touch is a far sense, like vision and hearing, contrary to our tendency to assume that it requires direct impression on the skin by an object. Vibration of the earth (felt in our feet) may signal the imminent approach of a train or a herd of wild buffalo. In a real sense, a tool becomes an extension of one’s body; the sensory site moves out to the tool tip. These comments further underline the claim that understanding haptics has important implications for the effective use and design of tools.

Arguably, Katz’s most important contribution to haptics research was on the subject of active and passive touch. When a subject is allowed to independently direct the movements of her hand, she is able to make a much more detailed report of surface texture than when the object is moved under her passive fingertips. Rather boldly, and with much foresight, Katz proposed an altogether different kind of organ for the sense of touch: the hand. By identifying the hand as the seat of haptic perception, he emphasized the role of intentional movement. He essentially coupled the performatory function of the hand to its perceptual function. By naming an organ that includes muscles, joints, and skin, he coupled the kinesthetic sense to the tactile. In certain instances, he claimed, two hands may be considered the organ of touch just as two eyes may be considered the organ of vision.

Geza Revesz (discussed in [27]) was particularly interested in the development of haptic perception in the blind and especially the coding of spatial information. According to Revesz, haptic recognition of objects is not immediate, as it is in vision, but requires constructive processing of sequentially acquired information. In haptics, the construction of the whole is a cognitive process that follows perception of parts. Revesz emphasized the spatial nature of haptics and its possibilities for apprehending an object from all sides. His theories and experiments with blind persons have had important implications for the development of aids for the blind, such as tactile maps. Perspective cues, occlusion, and background fading, each of which work so well in drawings presented to the eyes, do not work well in raised-line drawings presented to the hands. Recognition of three-dimensional models of objects with the hands, in contrast, is very good.

Gibson [19] contributed in subtle but important ways to the field of psychophysics and to haptics in particular. Gibson was interested in fostering a more ecological approach to research in sensory processes and perception, an approach that takes into account all properties of an environment that may have relevance to a person with particular intentions within that environment. He argued that perceptual psychologists should study recognition of objects rather than such “intellectual” processes as memory or imagination, or such low-level phenomena as stimulus response. Gibson proposed that perception is not simply a process of information gathering by the senses and subsequent processing by perceptual centers, but the result of a hierarchical perceptual system whose function depends on active participation by the perceiver. For example, the visual system includes not only the eyes and visual cortex but also the active
eye muscles, the actively positioned head, and even the mobile body. The haptic system, in addition to the tactile and kinesthetic sensors and somatosensory cortex, includes the active muscles of the arms, hands, and fingers.

Gibson, like Katz, stressed the importance of intentional movement in haptic perception. He preferred to think of active touch as a separate sense. Even when a subject has no intention of manipulating an object, she will choose to run her fingers over the object when left to her own devices. Certainly the fingertips are to the haptic sense as the fovea is to the visual sense: an area with a high concentration of sensors, and thus particular acuity. The fingers may be moved to place the highest concentration of sensors on the area of interest. Movement may be used to produce vibration and transient stimuli, which we know to be important from the experiments of Katz.

Gibson pointed to yet another reason for exploratory movement of the hand: to “isolate invariants” in the flux of incoming sensory information. Just as the image of an object maintains identity as it moves across the retina, or the sound of an instrument maintains identity as its changing pitch moves the stimulus across the basilar membrane, so an object maintains its identity as its depression moves across the skin. The identity even persists as the object is moved to less sensitive areas of the arm, and it is felt to maintain a fixed position in space as the arm glides by it. These facts, central to Gestalt theory, were underlined by Gibson and used as a further basis for understanding active touch. The exploratory movements are used to produce known changes in the stimulus flux while monitoring patterns that remain self-consistent. Thus, active touch is used to test object identity hypotheses, in Gibson’s words, to “isolate the invariants.”

Gibson also demonstrated that a subject passively presented with a haptic stimulus will describe an object in subjective terms, noting the sensations on the hand. By contrast, a subject who is allowed to explore actively will tend to report object properties and object identity. Under active exploration, she will tend to externalize the object or ascribe percepts to the object in the external world. For example, when a violin bow is placed on the palm of a subject’s passive hand, she will report the sensations of contact on the skin, whereas a subject who is allowed to actively explore will readily identify the object and report object properties rather than describe sensations. Furthermore, when a string is bowed, the contact is experienced at the bow hairs and not in the hand.

Today the field of haptics has many proponents in academe and industry. From our present vantage point in history, we can identify reasons for the earlier lack of research interest in haptics. Certainly the haptic senses are more complex than the auditory or the visual, in that their function is coupled to movement and active participation by the subject. And further, the availability of an experimental apparatus for psychophysical study in haptics, the haptic interface, has been lacking until now.

Many open questions remain in haptics. We are still not sure if we have an answer to the question that Aristotle raised: What is to haptics as sound is to hearing and color is to seeing? As is apparent from experiments with active and passive touch, the notion of haptic sensation cannot be divorced from the notion of manipulation. Furthermore, the spatial and temporal sensitivity of the haptic sensors is not fully understood. Much research, especially using haptic interfaces, will likely lead to new results. As never before, psychologists and mechanical engineers are collaborating to understand human haptic perception. Results in the field have important implications for virtual reality: the effective design of virtual objects that can be touched through a haptic interface requires a thorough understanding of what is salient to the haptic senses.

### 23.3.3 Anticipatory Control

In the manipulation of objects that are familiar, one can predict the response to a given input. Thus, there is no need to use feedback control. One can use control without feedback, called open loop control or anticipatory control, wherein one anticipates an object’s response to a given manipulation. From among many possible manipulations, one is chosen that is expected to give the desired response, according to one’s best knowledge or guess. Anticipatory control is closely related to ballistic control. However, in, ballistic control, not only the feedback, but also the feedforward path, is cut before the manipulation task is complete. In the case of a ball aimed at a target and thrown, for instance, one has no control over the ball’s trajectory...
after it leaves the hand. Both anticipatory control and ballistic control require that the human be familiar with the behavior of an environmental object or system, under a variety of input or manipulation strategies.

An object that is unfamiliar might be classifiable into a group of similar objects. If it is similar to another object, say in terms of size, simple scaling of the candidate manipulation/response interaction set (or model) would probably be appropriate. Scaling of an already available candidate input/output pair can be considered a case of using an internal model. Evidence for the existence of internal models is found in a manipulation experience with which most people are familiar firsthand. When grasping and lifting a suitcase, one may be surprised to have the suitcase rise off the table faster and higher than expected, only to realize that the suitcase is empty. The grasp and lift strategy was chosen and planned for a heavy suitcase.

The most important reason that humans and animals use anticipatory control rather than feedback control is that adequate time is not available for feedback control. There are significant delays in the arrival of sensory information, partly because of the slow (compared with electric wires) conduction of impulses along nerve fibers. Likewise, the execution of commands by muscles is accompanied by delays. Human response times, which include both sensory delays and command delays (the subject registers perception by pushing a button) are on the order of 180 msec. Yet many musical events, for example, are faster. A fast trill is 8 to 10 Hz. With less than 100 msec per cycle, the musician must issue commands almost a full cycle ahead of execution. Actuation based on a comparison between desired and feedback information is not possible.

### 23.4 Haptic Interface

As any person working with haptic interface technology soon finds out, sustained oscillations often hinder the display of virtual environments, especially hard virtual surfaces. Even the simplest virtual wall, whose physical counterpart is passive and well behaved, may nonetheless exhibit chatter when rendered through a haptic interface that is grasped by the user in a certain way. Suppose the following algorithm is used to render a virtual wall:

\[
F_e = \begin{cases} 
  k(x_e - x_o), & x_e \geq x_o \\
  0, & x_e < x_o 
\end{cases}
\]  

(23.1)

where \(k\) is a spring constant and \(x_o\) is a boundary locating the wall. Using this simple algorithm, and even when the haptic interface is equipped with the most ideal collocated sensors and actuators and lacks significant structural dynamics, one will notice that chatter appears. Chatter, or sustained oscillations often involving repeated contact with the wall, reflects a limit cycle in the coupled dynamics of human, haptic interface, and virtual environment. As the spring stiffness \(k\) is increased, the chatter will intensify. Somehow, energy not accounted for in the models of the virtual environment and interface and not provided by the user is leaking into the system to produce the oscillatory behavior. Typically, the frequency of oscillation is higher than the capabilities of human voluntary movement or even involuntary tremor, indicating that the energy required to sustain the oscillations is supplied by the interface/virtual environment system.

Origins of the energy leak include a half-sample delay associated with the zero-order-hold that intervenes between the discrete virtual environment and the haptic interface hardware. Other possible culprits include computational delay, signal quantization imposed by sensors, and delays associated with filtering or numerical differentiation to produce velocity estimates.

A simple graphical explanation of the energy-instilling effects of sampling on the virtual wall is available in Figure 23.6. For simplicity, the wall has been located at the origin: \(x_o = 0\). Overlayed on the spring law \(F_e = kx_e\) is a trace describing the reaction force of a wall generated using Equation (23.1) that was penetrated a single time to depth \(x_{max}\). The trace has a staircase shape because \(x_e\) is subject to sampling and the force \(F_e\) is zero-order held. As \(x_e\) increases, the held reaction force is less than that given by the continuous spring law, except at the sampling times, where \(F_e = kx_e\). Then as \(x_e\) decreases again, the held force is greater than the spring law, except at the sampling times. As a result, the work done on the wall...
Figure 23.6 leads naturally to the idea of predicting the displacement \( x_e \) a half-sample ahead to compensate for the effects of the zero-order-hold. The idea is to use a staircase approximation to the spring law that preserves area under the curve or equivalently preserves the force-displacement integral. The average of \( x_e(n) \) and a value \( \hat{x}_e(n + 1) \) predicted a full sample ahead may be used as a half-sample prediction of \( x_e \), yielding the algorithm

\[
F_e(n) = k \left( x_e(n) + \hat{x}_e(n + 1) \right)
\]  

(23.2)

If the velocity \( \dot{x}_e \) is available, the prediction \( \hat{x}_e(n + 1) \) may be obtained using \( \hat{x}_e(n + 1) \approx x_e(n) + \dot{x}_e(n)T \), which yields the algorithm

\[
F_e(n) = k x_e(n) + \frac{k T}{2} \dot{x}_e(n)
\]  

(23.3)

The second term can be interpreted as a virtual damper that is placed in parallel with the virtual spring to dissipate the energy generated by the zero-order-hold.

Very often the velocity \( \dot{x}_e \) is not available from a sensor and must instead be estimated based on sampled \( x_e \) measurements. If a backward difference is used, \( \dot{x}_e(n) \approx \frac{1}{T}(x_e(n) - x_e(n - 1)) \) which produces

\[
\dot{x}_e(n + 1) = 2x(n) - x(n - 1)
\]  

(23.4)

When substituted into Equation (23.3), this rule produces the following spring law:

\[
F_e(n) = k(1.5x(n) - 0.5x(n - 1))
\]  

(23.5)
A virtual wall rendered using this spring law will work significantly better than the wall rendered using a law without prediction, Equation (23.1). But it may be even further improved. Ellis [28] borrowed the idea of prediction–correction from numerical methods to further improve Equation (23.5). At sample \( n \) the displacement \( x_e(n) \) is available from a sensor and may be compared with the previously estimated \( \hat{x}_e(n) \). The error incurred as a result of using \( \hat{x}_e \) in place of \( x_e \) may be corrected. The difference between the actual \( x_e(n) \) and the previously predicted value \( \hat{x}_e(n) \) is added as a corrector term, yielding the predictor-corrector algorithm

\[
F_e(n) = \frac{k}{2} (2x_e(n) + \hat{x}_e(n + 1) - \hat{x}_e(n))
\] (23.6)

Substituting for the predicted values using Equation (23.4), the following algorithm is produced:

\[
F_e(n) = k(2x(n) - 1.5x(n - 1) + 0.5x(n - 2))
\] (23.7)

This algorithm can be extended beyond the virtual wall and applied more generally as a filter to the force to be displayed by the haptic interface [28]. Taking \( \tilde{F}_e \) as the filtered force and \( F_e \) has the name of the constitutive law of the virtual environment,

\[
\tilde{F}_e(n) = 2F_e(n) - 1.5F_e(n - 1) + 0.5F_e(n - 2)
\] (23.8)

### 23.4.1 Compensation Based on System Models

The prediction Equation (23.4) used in the algorithms above is based on the forward-Euler rule and a first-difference estimate of velocity. It uses a polynomial fit to sensor data from the present and recent past to predict sensor readings slightly into the future and, thus, account for delays. An alternative approach is to model the coupled dynamical system, including the virtual environment, haptic interface, and the human user, and use simulation or analytical solutions to the model to predict future states. This is the approach adopted in [29]. If the action of the human is modeled as a constant force and the human biomechanics as a second order system, the entire coupled system, including virtual wall and inertia and damping in the haptic interface, can be modeled as a lumped second order system with switching dynamics. Justification for the simple models of the human user are drawn from the disparate time-scales of sampling and volitional control, and the numerous demonstrations of good fit between second order models and experimental characterizations of human biomechanics [30]. Analytical solutions of the second order dynamics may be used to predict states between sample times. A half-sample prediction may be used to compensate for the effects of the zero-order hold.

Because the instants at which the wall threshold is crossed generally fall between sampling times, the effect of the zero-order-hold cannot be compensated with a half-sample prediction for those sampling periods that span the threshold crossings. However, the analytic solution may be used to account precisely for the change in dynamics (wall on/wall off) that occurs at the threshold crossings. The solution may be used as a goal state for a dead-beat controller that then drives the sampled data dynamics to the goal in two steps. The dead-beat controller is invoked at each threshold crossing and complements the half-sample prediction to produce sampled data system simulation which fully emulates the continuous system dynamics in [29].

### 23.4.2 Passivity Applied to Haptic Interface

An alternative approach to the analysis of stability in the coupled dynamics of the human/interface/virtual environment is based on the passivity theorem. When a virtual environment is modelled after a passive physical system, yet its rendering through a haptic interface exhibits sustained oscillations, analytical treatments like passivity, involving the production and dissipation of energy naturally come to mind. Another attractive feature of the passivity approach is that to characterize a system as passive, only an
input/output description is necessary. A full state-space model is not needed. Rather, system components can be handled as members of classes with certain properties, and results are significantly more modular in nature. Note that passivity is a property of the virtual environment alone; it is not a system property like stability. Components whose input-output properties are the same but internal realizations differ may be substituted for one another. Also, time delays can be accommodated, whether arising from sampling, computational, or communication delay, or having to do with explicit integration schemes.

By the passivity theorem, if the human and virtual environment are assumed passive and the haptic interface assumed strictly passive (since it certainly has some energy dissipation due to friction and damping), then the coupled system will be stable. The contra-positive statement says that instability of the interconnected system implies that one of the three system components is not passive. Because the haptic interface is constructed of physical components, it cannot be the culprit. At first it would not seem so reasonable to assume that a human operator is a passive system, since, of course, humans metabolize food to produce mechanical energy. But because the unstable, chatter-like behavior so often observed in haptic rendering is typically characterized by frequencies that lie outside the range of human motor capabilities, one typically assumes that the human is in fact passive. Thus, the virtual environment is implicated, and measures to ensure its passivity should be taken to remedy the instability.

A system with input $u(t) \in \mathbb{R}^n$ and output $y(t) \in \mathbb{R}^n$ is passive if there exists a nonnegative function $W(x)$, called a storage function, such that

$$W(x(t)) \leq W(x(0)) + \int_0^t y(\tau)^T u(\tau) d\tau$$

(23.9)

for all inputs $u(t)$ and all $t \geq 0$. The term $W(x(0))$ is the initial stored energy in the system and the product $y^T u$ is called the supply rate for the system. If $y$ and $u$ are force and velocity, then the integral of the supply rate is the power supplied to the system up to time $t$. The stipulation that the inequality must hold for any input $u(t)$ means that no matter how the system is driven, its net absorbed energy will always exceed its initial stored energy. Parseval’s theorem can be used to transform the passivity definitions above into expressions in the frequency domain. For linear systems, Parseval’s theorem shows that passivity corresponds to strict positive realness.

Colgate and Schenkel [31] determined parameter values that guarantee passivity of the virtual wall of Equation (23.1) when it is rendered through a zero-order-hold. Colgate’s analysis is based on an application of the small gain theorem (as is the passivity theorem itself). But because the small gain theorem takes only magnitude information into account and completely disregards phase, a linear fractional transformation (LFT) (which has equivalent interpretations as a loop transformation and a coordinate change) must be used to reduce conservativeness. Using only the constraint that the human operator be passive, Colgate first finds the area in the Nyquist plane within which a passive human operator in feedback connection with the haptic interface and linked with a zero-order hold and integrator must lie. This area (a disk) can be mapped to the unit disk (uncertain phase; unity magnitude) by an LFT. A corresponding LFT is found for the discrete controller (the virtual wall) in [32]. The unit disk then becomes a bound in the Nyquist plane for the mapped discrete controller that, if satisfied, guarantees coupled stability by the small gain theorem. The result may be stated simply as

$$b > \frac{KT}{2} + |B|$$

(23.10)

where $b$ is viscous damping in the physical haptic device, $K$ is the virtual wall stiffness, $T$ is the sampling time, and $B$ characterizes a virtual viscous damper in parallel with the virtual spring. The result is both necessary and sufficient to guarantee passivity for the constitutive law without the unilateral nonlinearity, and sufficient for the virtual wall. This equation says that excess damping $b$ in the haptic interface device can be used to account for a deficiency of passivity in the virtual environment (due to half-sample delay in the zero-order-hold). Note the term $KT/2$ which appeared above in Equation (23.3) as the damping coefficient of a virtual damper that, when added to the spring law, compensated for the effects of the...
zero-order-hold. Here, physical damping in the haptic interface, rather than virtual damping, is used to account for the effects of the zero-order hold.

Equation (23.10) can be interpreted either as a design directive or as an insight. As a directive, Equation (23.10) says that an excess of dissipativity should be built into the haptic device and then its effect compensated by negative virtual damping \( B \) (note the absolute value around \( B \)). In effect, a physical damper \( b \) extracts energy at all frequencies (including the high frequencies above the Nyquist frequency) whereas a negative virtual damper only adds energy at low frequencies (where its effect is felt). Thus, in the end, stability is guaranteed and transparency (wallness) is not compromised. However, viscous dampers (at least reliable, linear ones) are not readily available commercially and are rather difficult to build and maintain.

Adams and Hannaford [12] and also [33] applied a passivity argument to a broader class of virtual environments than the Colgate papers, including general differential ODE models. In addition, the Adams and Hannaford framework brought in the virtual coupler and extended to impedance and admittance implementations. However, their work does not address the excess of passivity of the device that may be used to account for nonpassive behavior of the virtual environment. The virtual environment is, by necessity, nonpassive since explicit integrators must be used in real-time simulation.

Miller et al. [34] extend the passivity arguments of [12] and [31] to include concepts of input-strict and output-strict passivity. Using these tools, an excess of passivity in the haptic device can be used explicitly to account for a lack of passivity in the virtual environment. The results apply to virtual environments with nonlinearities and even time delays.

Hannaford and Ryu in [35] present a general passivity controller and a passivity observer that work together to guarantee that the haptic rendering is passive. The strength of this compensator is its generality; it does not depend on any knowledge of the system driving the interface nor on the behavior of the user. The generality of the solution is also a weakness though. If the human operator is active for a long period of time, then the system driving the interface can be active for a long period of time and exhibit undesirable active interactions with the operator.

23.5 Virtual Environment

A set of surface patches and their interconnection can be used to describe the geometry of each body in the virtual environment. The whole collection of surface patches along with their connected graphs is called the geometric model. Computations involving the geometric model, in a process called collision detection, identify points in time at which bodies within the virtual environment make contact with one another. At or during times of contact, the collision detector triggers another process called the interaction calculator, as shown in the flowchart of Figure 23.7. The interaction calculator uses an interaction model to compute the appropriate impulse response or interaction force between bodies. A third process is the solution of the forward dynamic model for the collection of bodies, producing motion in response to the applied forces and moments. The applied forces and moments include both those applied by the virtual coupler and the results of the interaction calculator.

The interaction calculator passes either new initial conditions (the result of an impulse response computation) or interaction forces (the result of continuous contact) to the forward dynamics solver. Equal and opposite forces and moments are applied to pairs of interacting bodies. The forward dynamics solver operates on the forward dynamics model or equations of motion, which is a set of differential equations in the configuration and motion variables and inertia parameters. The dynamic model might also contain embedded holonomic or nonholonomic constraint equations, written in terms of certain geometric parameters that are not necessarily part of the geometric model.

The use of a collision detector and interaction calculator as in Figure 23.7 ensures that interacting bodies respond to each other’s presence. Note that the interaction calculator is called upon only intermittently whereas the collision detector and forward dynamics solver run continually, either alongside or subsequent to each other in computational time.

Let us now consider in turn each of the elements of the flow chart in greater detail.
23.5.1 Collision Detector

To calculate a global solution in a computationally efficient manner, it is very common to handle the collision detection problem in two parts: a broad phase, which involves a coarse global search for potentially interacting surfaces, and a narrow phase, which is usually based on a fast local optimization scheme on convex surface patches. To handle nonconvex shapes, preprocessing is performed to decompose them into sets of convex surface patches. However, there exist shapes for which such a decomposition is not possible, for example, a torus.

23.5.1.1 Broad Phase

The broad phase is composed of two major steps. First, a global proximity test is performed using hierarchies of bounding volumes or spatial decompositions for each surface patch. Among the most widely used bounding volumes and space decompositions are the octrees [36], k-d trees [37], BSP-trees [38], axis-aligned bounding boxes (AABBs) [39], and oriented bounding boxes (OBBs) [40].

During the global proximity test, the distances between bounding boxes for each pair of surface patches drawn from all pairs of bodies are compared with a threshold distance and surfaces that are too distant to be contacting are pruned away. Remaining surfaces are set to be active.

In the second step of the broad phase, approximate interaction points on active surfaces are calculated. For example, if the geometric models are represented by non-uniform rational B-splines (NURBS), control polygons of the active surface models can be used to calculate a first order approximation to the closest points on these surfaces. Specifically, bounding box centroids of each interacting surface patch can be projected onto the polygonal control mesh of the other patch. Using the strong convex hull property of
NURBS surfaces, one can determine the candidate span and interpolate the surface parameters from the node values. These approximate projections serve as good initialization points for the narrow phase of the collision detector.

### 23.5.1.2 Narrow Phase

After initialization with the approximations calculated in the broad phase, the narrow phase employs an algorithm to iteratively update the minimum distance between each active pair of surfaces.

Previous work in narrow phase collision detection has concentrated on computing the minimum distance between convex polyhedral objects. State-of-the-art algorithms for computing the distance between convex polyhedra are based on the algorithm by Gilbert, Johnson, and Keerthi (GJK) [41, 42], the algorithm of Lin and Canny [43], and the algorithm by Mirtich [44].

The GJK algorithm makes use of Minkowski difference and (simplex-based) convex optimization techniques to calculate the minimum distance. The iterative algorithm generates a sequence of ever improving intermediate steps within the polyhedra to converge to the true solution. The algorithm of Lin and Canny makes use of Voronoi regions and temporal/spatial coherence between successive queries to navigate along the boundaries of the polyhedra in the direction of decreasing distance. The V-Clip algorithm by Mirtich is reminiscent of the Lin and Canny closest features algorithm but makes several improvements.

Less literature exists on direct collision detection for nonpolygonal models. Gilbert et al. extended their algorithm to general convex objects in [45]. In a related paper [46], Turnbull and Cameron modify the widely used GJK algorithm to handle convex shapes defined using NURBS. Similarly, in [47, 48] Lin and Manocha present an algorithm for curved models composed of spline or algebraic surfaces by extending their earlier algorithm for polyhedra.

Finally, a new class of algorithms, namely, minimum distance tracking algorithms for parametric surfaces, is presented in [49–52]. These algorithms are designed to maintain the minimum distance between two parametric surfaces by integrating the differential kinematics of the surfaces as the surfaces undergo rigid body motion.

### 23.5.2 Interaction Calculator

The interaction calculator, triggered into action by the collision detector, computes inputs (forces and moments) and new initial conditions (in the case of impulse response) for the forward dynamics solver. In so doing, it realizes interaction between bodies, whether such interaction occurs during a finite or an infinitesimal period of time. As indicated in Figure 23.7, the interaction calculator calls upon an interaction model and the geometric model to compute its response. If impulse response is also used, the interaction calculator additionally calls upon portions of the dynamic model.

The most widely used interaction model in both haptic rendering and multibody dynamics is called the penalty method: it assumes compliant contact characterized by a spring-damper pair. The force response is then proportional to the amount and rate of interpenetration. The interaction calculator need only query the geometric model using these surface parameters to determine the interpenetration vector. To compute a replacement (a resultant applied at the center of mass and a couple applied to the body) for the set of all contact forces and moments acting on a body for use as input to the dynamic model, the interaction calculator queries the geometric model using the surface parameters and then carries out the appropriate dot and cross products and vector sums. Using replacements for the set of interaction forces allows the dynamic model to be fully divorced from the geometric model.

The penalty contact model allows interpenetration between the objects within the virtual environment, which in a certain sense is nonphysical. However, the penalty method also eliminates the need for impulse response models, since all interactions take finite time. Important alternatives to the penalty method include the direct computation of constraint forces using the solution of a linear complementarity problem espoused by Baraff [53, 54] and widely adopted within the computer graphics community. The formulation by Mirtich and Canny [55] uses only impulses to account for interaction, with many tiny repeated
impulses for the case of sustained contact. Note that computation of impulses requires that the interaction calculator query the dynamic model with the current extremal point parameters and system configuration to determine the effective mass and inertia at each contact point.

It is important to carefully distinguish between the role of the penalty method and the virtual coupler. The virtual coupler serves as a filter between the virtual environment and haptic device whose design mitigates instability and depends on the energetic properties of the closed-loop system components. The penalty method, on the other hand, though it may also be modelled by a spring-damper coupler, is an empirical law chosen to produce reasonable behavior in multibody system simulation. Parameter values used in the penalty method also have implications for stability, though in this case it is not haptic rendering system but rather numerical stability that is at issue. Differential equation stiffness is, of course, also determined by penalty method parameter values.

Associated with the penalty method is another subtle issue, that of requiring a unique solution to the maximum distance problem when the interpenetrated portions of two bodies are not both strictly convex or there exists a medial axis [56] within the intersection. This is the issue which the god-object or proxy methods address [57, 59].

### 23.5.3 Forward Dynamics Solver

The final component comprising the haptics-equipped simulator is a differential equation solver that advances the solution of the equations of motion in real time. The equations of motion are a set of differential equations constructed by applying a principal of mechanics to kinematic, inertial, and/or energy expressions derived from a description of the virtual environment in terms of configuration and motion variables (generalized coordinates and their derivatives) and mass distribution properties. Typically the virtual environment is described as a set of rigid bodies and multibody systems interconnected by springs, dampers, and joints. In such case the equations of motion become ordinary differential equations (ODEs), expressing time-derivatives of generalized coordinates as functions of contact or distance forces and moments acting between bodies of the virtual environment. In Figure 23.4, representative bodies A, B, and P comprise the virtual environment, while the forces and moments in the springs and dampers that make up the virtual coupler between bodies P and E are inputs or arguments to the equations of motion. The configuration (expressed by the generalized coordinates) and motion (expressed by the generalized coordinate derivatives) of bodies A, B, and P are then computed by solving the equations of motion (see also Figure 23.5). One may also say that the state (configuration and motion) of the virtual environment is advanced in time by the ODE solver.

Meanwhile, the collision detector runs in parallel with the solution of the equations of motion, monitoring the motion of bodies A, B, and P and occasionally triggering the interaction calculator. The interaction calculator runs between time-steps and passes its results (impulses and forces) to the equations of motion for continued simulation.

Quite often the virtual environment is modeled as a constrained multibody system, and expressed as a set of differential equations accompanied by algebraic constraint equations. For example, constraint appending using the method of Lagrange Multipliers produces differential-algebraic equations (DAEs). In such case, a DAE solver is needed for simulation. Note that DAE solvers are not generally engineered for use in real-time or with constant step-size and usually require stabilization. Alternatively, constrained multibody systems may be formulated as ODEs and then simulated using standard ODE solvers using constraint-embedding techniques. Constraint embedding can take place symbolically (usually undertaken prior to simulation time) or numerically (possibly undertaken during simulation and in response to runtime events).

Alternatives to the forward dynamics/virtual coupler formulation described above have been developed for tying together the dynamic model and the haptic interface. For example, the configuration and motion of the haptic device image (body E in Figure 23.5) might be driven by sensors on the haptic interface. A constraint equation can then be used to impose that configuration and motion on the dynamic model of the virtual environment.
23.6 Concluding Remarks

In this chapter, we have presented a framework for rendering virtual objects for the sense of touch. Using haptic interface technology, virtual objects can not only be seen, but felt and manipulated. The haptic interface is a robotic device that intervenes between a human user and a simulation engine, to create, under control of the simulation engine, an appropriate mechanical response to the mechanical excitation imposed by the human user. ‘Appropriate’ here is judged according to the closeness of the haptic interface mechanical response to that of the target virtual environment. If the excitation from the user is considered to be motion, the response is force and vice-versa. Fundamentally, the haptic interface is a multi-input multi-output system. From a control engineering perspective, the haptic interface is a plant simultaneously actuated and sensed by two controllers: the human user and the simulation engine. Thus the behavior of the haptic interface is subject to the influence of the human user, the virtual environment simulator, and finally its own mechanics. As such, its ultimate performance requires careful consideration of the capabilities and requirements of all three entities.

While robotics technology strives continually to instill intelligence into robots so that they may act autonomously, haptic interface technology strives to strike all intelligence out of the haptic interface—to get out of the way of the human user. After all, the human user is interested not in interacting with the haptic interface, but with the virtual environment. The user wants to retain all intelligence and authority. As it turns out, to get out of the way presents some of the most challenging design and analysis problems yet posed in the greater field of robotics.

References


