CENTER FOR COMPUTER RESEARCH IN MUSIC AND ACOUSTICS
JANUARY 1996

DEPARTMENT OF MUSIC
REPORT NO. STAN-M-92

HAPTIC DISPLAY OF SYSTEMS WITH CHANGING KINEMATIC CONSTRAINTS: THE VIRTUAL PIANO ACTION

RICHARD BRENT GILLESPIE

RESEARCH SPONSORED BY
THE CCRMA INDUSTRIAL AFFILIATE PROGRAM

CCRMA
DEPARTMENT OF MUSIC
STANFORD UNIVERSITY
STANFORD, CALIFORNIA 94305-8180

HAPTIC DISPLAY OF SYSTEMS WITH CHANGING KINEMATIC CONSTRAINTS: THE VIRTUAL PIANO ACTION

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF MECHANICAL ENGINEERING
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

By
Richard Brent Gillespie
January 1996

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Mark R. Cutkosky
(Principal Adviser)

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

T. R. Kane

Thomas R. Kane

I certify that I have read this dissertation and that in my opinion it is fully adequate, in scope and in quality, as a dissertation for the degree of Doctor of Philosophy.

Stephen Boyd

Approved for the University Committee on Graduate Studies:

Abstract

Skilled keyboardists enjoy very fine control over musical sounds at the piano but must make do without that fine control at the synthesizer keyboard. In contrast to the piano action, the typical synthesizer action is not subject to changing kinematic constraints and is therefore missing transient features in its mechanical impedance (touch-response). Keyboardists rely on such haptic features to develop and execute the aforementioned fine control. To remedy the inferior utility of synthesizer keyboards, this thesis develops and applies haptic interface technology. Each synthesizer key is motorized and a multibody, variable structure dynamical model of the piano action is simulated in real-time in a human-in-the-loop scheme to re-create the response and the varying mechanical impedance of the piano action. A combined simulation and experimental apparatus comprising a seven key motorized keyboard is described. For use as a haptic interface control engine, a detailed dynamical model of the piano action is developed using Kane's method. Computationally efficient submodels are constructed for the piano action in each of its constraint conditions. Simulation schemes based on a finite state machine are developed so that the submodels may be interactively sequenced together.

Limitations to the fidelity of haptic rendering invariably arise when the simulator is implemented as a sampled data controller. Restrictions must be placed on the mechanical impedance of the virtual object or exceptional computational power must be demanded of the interface controller lest meddlesome chatter arise between user and virtual object. This work notes that the destabilizing effects of sampled-data and computational delays can be fully compensated out if the entire coupled dynamical system is modeled: interface device and human limb. Such methods are fully explored for the virtual wall, the simplest virtual object subject to changing constraints. New algorithms are presented for the virtual wall which address the destabilizing effects of discrete control and discontinuous control, yet account for the coupled-in dynamics of the human. Specifically, the deleterious effects of the sample and hold operator and the asynchrony of constraint threshold crossings with sampling times are eliminated. Model-based prediction, digital control design techniques, and deadbeat control are employed.

Acknowledgements

I would like to thank my advisor Mark Cutkosky for his generous contributions of guidance, know-how, and perspective. I have stretched and grown under Mark's always right-on-target tutelage. I owe much to Thomas Kane and Stephen Boyd, who have introduced me to many powerful tools in dynamics and controls and have enthusiastically counseled me in their application. I have also profited from numerous in-depth discussions with the other members of my committee, Arthur Bryson and George Barth. Each of my committee members have seriously enriched my Stanford experience. I will always draw inspiration from them.

I am grateful to Ed Colgate for his thoughtful and extremely valuable review of Chapters 5 and 6. I am really looking forward to working with Ed in my postdoc at Northwestern.

I would like to thank John Chowning, Chris Chafe, Julius Smith, George Barth, and Max Mathews of the Music Department for creating and embodying good CCRMA. I will miss CCRMA, its freedom and spirit and sounds. CCRMA has given me the opportunity to combine my interests in music and engineering, which simply would not have been possible at any other institution. I am very lucky.

I have enjoyed waxing technical with my fellow graduate students at CCRMA, especially Perry, Scott, and Bill. Working with Sile has been and will continue to be a true pleasure. Thanks to my cohorts at CDR: Marc, Dean, and Jim for sharing ideas and wares. I am very glad to have fellow aspiring haptics gurus Paul Millman and Karon MacLean at Northwestern and MIT, now Harvard and Interval. I look forward to continued collaboration with you excellent people!

I thank the folks at Interval Research Corp. and Immersion Corp., especially Bill Verplank and Louis Rosenberg, for generating and enabling some wonderful applications in this exciting new field called computational haptics.

My piano teachers and coaches over the years, Verna Randal, Lois Brandwynne, Mack McCray, Paul Hersch, and George Barth have provided inspiration and encouragement in matters of music *and* engineering. Many parts of this thesis were written with them in mind.

I thank Walt for his fine editorial services and his brotherly reverence and irreverence, always well-timed. Finally, I thank my parents for their unending love and support. They know my gratitude.

The CCRMA industrial affiliates program has provided the financial support for this work. In addition, Yamaha Corp. has donated a piano action model, Delta Tau Corp., an eight-axis motion controller. Dr. Jim Walton performed the analyses of the experimental data appearing in Chapter 3, and Professor Tom Bowman arranged for a loan of the high speed video equipment used to collect that data.

To my parents.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

The work of this thesis has been kindled by the desire for a certain unique product—an electronic keyboard instrument which responds, both in terms of sound and feel, just like an acoustic grand piano. Keyboardists familiar with the piano and the synthesizer have long been dreaming of one instrument with the advantages of both: the response and accompanying expressive potential of the piano and the programmability of the synthesizer. Certainly we are not the first to identify this product need —synthesizer manufacturers have been clamoring for a decade to come up with an electronic version of the piano. Evidently there remain many challenges in the design of such an instrument since by the standards of any pianist, existing electronic pianos are poor substitutes for acoustic pianos.

To produce a device which emulates the behavior of a proven predecessor seems like a simpler task than starting from scratch and without guidelines, especially given the many new technologies available to us which were not available to the designers of that predecessor. To emulate the mechanical and acoustical response of the piano in computer-driven hardware, however, has proven to be a very challenging task.

Theme for section 2.1

To begin, we are interested in the response of a mechanical system to some rather special driving inputs: those of a human. The responses in turn are to be judged by a human, and thus certain psychophysical factors will enter the discussion. Indeed, for its effectiveness as a musical instrument, the piano depends heavily on certain psychoacoustical and psychophysical phenomena. Simply put, the piano has survived (even flourished) by continually fooling the ears of its listeners and the hands of its players. The sounds it produces are strictly percussive, yet from these our ears somehow construct lyric melodies. The physics of its sound production are separated from its keyboard interface by a very complex system of levers, yet a player's fingers somehow find ways of manipulating all the available sound parameters. The deceptive habits of the piano will become the primary subject of Chapter 2. There I will undertake a thorough discussion of the psychophysics of the piano. My primary aim will be to extract design guidelines for a new digital instrument.

Roadmap to the document

Aiming to satisfy product desires defines the basic activities of almost every engineer, and thus this thesis touches upon a relatively broad range of engineering topics. Chapter 2 is somewhat self-contained, and should be accessible to all engineers and persons interested in computer music or electronic instrument making. Chapter 2 also introduces the work of the entire thesis. By itself, Chapter 2 is original only in that it is a rather complete collection of thoughts on the subject of keyboard instrument design. The actual results of our work are presented in the remaining chapters. Each of the sections in Chapter 2 introduces the work of a certain following chapter, as shown in Figure 1.1. Effectively, Chapter 2 is a launching pad for the rest of the document.

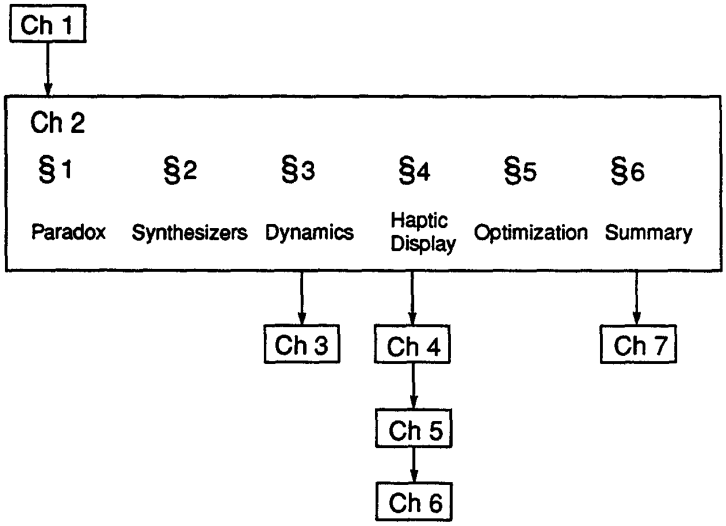


Figure 1.1: *Roadmap to the Document*

Theme for section 2.1

In section 2.1, the debate over just which sound parameters are actually under the control of a pianist will be taken up and for our purposes resolved. A brief literature review at the end of section 2.1 will establish that this debate has surfaced several times throughout the history of the piano.

Theme for section 2.2

Behind our interest in creating a digital instrument modeled after the acoustic piano lies the hope of landing one with a career like that which the acoustic piano has enjoyed. We further hope that our process of inventing a prosperous digital instrument will not take as long as the invention of the acoustic piano—which consumed two whole centuries! (approximately 1705-1900). The fact that the piano's design evolved over such a long period of time is due in part to the gradually developing technology to which it was tied, but also, quite undeniably, to its essence as a music-producing 'black box' filled with complex mechanisms whose purposes seem to border on subterfuge as discussed in section 2.1. We trust, however, that our understanding of the psychoacoustics and psychophysics of the piano will allow us to circumvent a few design iterations in our quest to produce a similar instrument. We even expect that a digital instrument will be able to take advantage of psychophysical factors in interesting new ways. In Chapter 2, section 2, I begin an analysis of the black box of the piano by defining its boundaries and the ports through which it interacts with its environment (the pianist and audience). A general definition of a musical instrument, which covers both acoustic and electronic versions, will be presented. Using this definition, the shortcomings of existing electronic pianos with regard to their potential as expressive instruments will be enumerated. Section 2.2 will then conclude with a critical look at the design principles which have been used in present-day commercial synthesizers and digital pianos, leaving the reader with a clear picture of where we, as instrument designers, need to start anew.

Theme for section 2.3 and Chapter 3

It will become necessary, before embarking on our piano emulator design effort, to study the piano and its dynamical behavior in detail. To this end, the process of modeling the grand piano action will begin in Chapter 2, section 3. We must fully understand the laws and mechanisms which govern the piano's behavior under a player's fingers if we are to duplicate that behavior. In particular, we will be interested in the feel or mechanical impedance of the piano at the key and will develop models of the piano action for the purposes of accounting for observed impedance characteristics. Simplified models will be presented in section 2.3. Section 2.3 will concentrate on the release and catch of the piano hammer by the remaining elements of the action. Our simplified models for this system will be reminiscent of a bouncing ball.

A mechanical system in which bodies may make and break contact with one another shall be called a system subject to ***changing kinematic constraints.*** The piano action is such a system, and its changing constraints are of particular interest because they give rise to features or mechanical impedance variations which can be felt (and in turn used or manipulated) at the key. Chapter 3 will present models for the piano action in each of its kinematic constraint conditions and further highlight the effects of changing constraints on the observed mechanical impedance at the key. A thorough literature review covering multibody model formulations of systems with changing kinematic constraints will be presented in Chapter 3. Our own model formulation, chosen for computational efficiency, will be compared and contrasted to other formulations.

Theme for section 2.4 and Chapter 4

In joining the digital piano design effort, we have decided to pay particular attention to the feel or 'touch-response' at the keyboard. Especially with regard to the feel, we believe that present-day digital pianos are gravely deficient. Our proposed method for re-creating, in a synthesizer keyboard, the feel of a grand piano or, for that matter, the feel of another keyboard instrument, is to motorize the keys and place them under computer control. This souped-up synthesizer will thus be touch-programmable—the feel of the keys and the relationship of the feel to the sound production parameters will be customizable or arbitrarily adjustable by the user. In section 4 of Chapter 2, I will discuss our approach to emulating the feel of a mechanical system such as the grand piano action with a motorized key under digital control. I will introduce the manner in which simulations of dynamical models of the piano action may be run in real-time in a human-in-the-loop scheme to re-create the feel of the action.

The portrayal of virtual touchable objects through a motorized device is a lively topic of research at present. Such a device is called a haptic interface or sometimes a haptic display. The word 'haptic' has been appropriated from the medical community, where it is used to refer in one word to the perceptual modalities of taction (senses of the skin), and kinesthesia (senses of the muscles). Our touch-programmable synthesizer keyboard can thus be considered an interface for virtual reality—it renders a virtual piano action or, at the touch of a button, a virtual harpsichord action, or perhaps the action of an altogether new instrument. I will contrast our approach with other possible approaches to mechanical system impedance emulation and briefly review the literature on this topic in section 2.4. Chapter 4 will introduce our unique design of a haptic interface and our techniques for the emulation of keyboard instruments. The manner in which the mechanical design of our haptic interface is related to the mechanical design of the piano action will be highlighted.

Theme for Chapter 5

A significant challenge which the entire haptic interface research community faces at present is the high-fidelity emulation of changing kinematic constraints. It turns out that even the very simplest of virtual objects containing changing kinematic constraints exhibit non-passive behavior—they tend to introduce energy into the coupled system made up of the device and the human, thereby causing sustained oscillations or 'chatter'. The prototypical simple object whose changing kinematic constraint tends to cause sustained oscillations is the virtual wall, especially a stiff virtual wall displayed with a slow sampling rate. These simulation difficulties are often simply labeled 'numerical problems' and swept under the rug, being relegated to that set of problems which will disappear when computers get faster. In haptic display, however, the challenge is not so easily forgotten. Interactive systems must be run in real-time, and the desire to share computer processing power with other operations such as graphical display will always be present. Therefore, we would like to ensure high-fidelity haptic display despite slow sampling rates.

Chapter 5 will describe the energetics of a haptic interface and its sampled-data controller attempting to emulate a wall, and will present improved controllers which do not exhibit sustained oscillations. These new controllers are immune to the destabilizing effects of two inescapable elements in any sampled-data implementation of a virtual wall: the zero order hold and the asynchrony of wall switching times with sampling times. Controller design techniques, which draw upon predictive simulation, digital domain design tools, and deadbeat control, will be fully developed.

Theme for Chapter 6

Chapter 6 will analyze the virtual wall controllers presented in Chapter 5. Measures for the energy introduced by the old standard controller designs will be sought so that the costs of implementing the new designs of Chapter 5 may be accurately weighed.

Chapters 5 and 6 treat difficulties which arise in the haptic display of a virtual piano action, namely the real-time rendering of changing kinematic constraints through a sampled data system. The treatment in Chapters 5 and 6, however, is quite narrow in that only a simpler, stand-in virtual system (the virtual wall) is considered. Most notably, dynamical models of the human finger coupled to the haptic interface are used in the improved virtual wall controller designs. Because the time-scale of the chatter problem is short (sustained oscillations run on the order of 10 to 50 Hz), and the need to account for volitional control on the part of the human is thereby obviated, successful virtual wall algorithms may be developed by taking into account the assumed mechanical properties of the human. In part to prepare for extending these results to longer time-scales, but also because there exists much interesting unchartered territory in the area, we will widen our viewpoint again in the remaining sections of Chapter 2.

Theme for section 2.6

To embark on a project whose primary aim is to improve upon an existing successful human interface design and to generalize that design by making it programmable, raises many interesting research questions. For example, what allows a pianist to maximize his control over the piano, and what role does the mechanical impedance of the piano play in that process? We have begun to address these questions with optimal control theory. Optimization theory allows us to treat the human/piano system in a framework which places each of the participants within this feedback system in an appropriate role. The human is modeled as an optimizing controller, attempting to maximize an output from the piano according to some objective function chosen for its musical significance. Our investigations in optimization will be outlined in section 2.6.

Finally, Chapter 7 will summarize and outline future work.

Chapter 2

Psychophysics of the Piano

> The moral for piano teachers is that so far as single notes are concerned, it does not matter how the pupil strikes the key, so long as he strikes it with the requisite degree of force. If this is right, the tone quality will be the same whether he strikes it with his fingers or even the end of his umbrella. As far as the scientist can see, that is all there is to the much debated problem on the piano touch.

–Sir James Jeans, from a lecture read to the English Piano Teachers' Association, Jan. 8, 1939 [52]

2.1 The piano: a paradox

The piano, when regarded both from the standpoint of an engineer and that of a musician, presents something of a paradox. The engineer, whose primary interests are learning from or improving upon the piano's design, will inevitably find it difficult to reconcile his convictions about this instrument with those of the musician, whose interests are centered around musical expression. The engineer points to the simple principles by which the piano produces sound and the correspondingly small set of controls over these principles which the piano makes available to its players at the keyboard. He underlines the fact that the piano is fundamentally a percussion instrument. The musician, on the other hand, points to the rich music which the piano can produce, nuanced not only in harmony and phrasing, but also in loudness and tone color. The expert pianist can even demonstrate independent control over each of these parameters. But given the degree of decoupling which the piano imposes between the pianist's fingers and its sound-producing mechanics, even the musician must concede that it is by no means apparent how one may use this instrument to produce evocative music.

In this introductory section, I will attempt to unravel the discussion between engineer and musician over the piano and the music it produces. I am interested in laying open this apparent paradox in order to motivate and enable the design of electronic musical instruments which more fully approach the piano in their expressive capabilities. If rapport could be established between engineer and pianist, the design of electronic instruments could in fact become a collaborative effort between the two, an activity which, arguably, has not yet occurred.

As already intimated, the investigations of this thesis center around a proposed re-design of electronic instruments, specifically, synthesizer keyboards. We envision a keyboard which offers the advantages of programmability, computerization, and mechanization, yet still realizes the full expressive capability of the grand piano. Not until we can re-engineer and modernize the piano without losing any of its capacity as an expressive instrument will we have invented a worthy successor in the form of a digital instrument.

A redesign of electronic pianos will naturally require that numerous design decisions be made; these decisions will in turn require quantitative bases. Thus, our analytical tools will be those of the engineer. Our endeavor, however, is completely accountable to and driven by the musician. Indeed, the extent to which this new instrument facilitates artistic expression for the musician will be the ultimate gauge of the success of a re-engineering of the piano. Our particular concern is the 'feel' of the piano at the keys, but before we address the topic of mechanical interactions between the pianist and piano, let us further motivate our redesign effort by revisiting the juxtaposition of views, the musician's and the engineer's.

2.1.1 Subdominant: The engineer's viewpoint

Compared to other musical instruments, the piano provides the musician with a very restricted set of controls over the parameters of sound. Unlike wind and bowed string instruments, which depend upon continuous excitation of a resonating body for the production of a tone, the piano offers no provisions for varying timbre [1] independent of intensity. For each pitch on the piano, there exists a one-to-one correspondence between intensity and timbre. The two are jointly determined by a key stroke. This fact has to do with the piano's percussive nature and will be further discussed below.

Unlike the voice (but similar to most other musical instruments), the piano is not capable of modulating the filtering properties of its resonating body. After the hammer-string interaction period is over, the sound of the piano evolves according to the static filtering properties of the string and soundboard.

[1] Timbre is a rather difficult term to define because it is a perceived quantity and further a multi-dimensioned quantity. The Acoustical Society of America has settled on defining timbre as *all tone qualities not already defined as pitch.* See [58]. I use the words timbre and tone color somewhat interchangeably to refer to the frequency spectrum of a tone.

The piano, along with its keyboard instrument relatives, does have one important advantage over other instruments: readily produced polyphony. It's immense reputation as an expressive instrument, however, cannot be fully accounted for by this one factor. The piano also enjoys status as an instrument of rich tone color. Its music is widely recognized as lyric, with a color palette practically on par with any string or wind instrument. Piano performances are characterized not only according to their range in intensity (loudness) but also according to their range in shading and timbre. In light of the percussive nature of the instrument, however, it really does come as a surprise that the piano should be revered for its ability to produce nuance in tone color. The phenomenon is especially astonishing since a pianist also lacks control over the precise pitches produced by the piano; these are under the control of the piano tuner. I will further illustrate the basis for the engineer's claim that the pianist has very limited control over the sound of the piano by considering the piano action as a kind of transform or map.

Mapping from gesture to sound parameters

Instigated by key-presses, sounds are produced on the piano by percussive hammer strikes on strings. The period of interaction between hammer and string, during which a waveform is set up on the string, is very brief (about 2 milliseconds) [3]. This waveform quickly evolves into a standing wave which gradually leaks its energy into the soundboard and from there into the surrounding air. The physics governing the hammer-string interaction are complex, owing in large part to the compression-hardening properties of the felt covering the hammer. For a look into the ongoing research of the hammer-string interaction, see [40] and [13]. Our particular interest, however, lies in the fact that the key and the pianist's finger are completely decoupled from the hammer during its brief period of interaction with the string. The hammer flies free of the jack which initially propels it some 2.5 milliseconds before striking the string [3]. The pianist has no means at his disposal for controlling the tone or the evolution of tone after the hammer has left the jack, except through the damper. Thus all parameters of the tone must be set up by the pianist before tone onset. [2]

The piano action, then, can be regarded as a system which maps a keypress, initiated at a

[2] The possibility of the hammer supporting more complex motions (such as vibration modes in the hammer shank) has been refuted by the excellent experimental studies of Anders Askenfelt and Jens Jannson [2]. Vibration modes in the hammer shank were hypothesized as a means for independent control over intensity and timbre. If the motion of the piano hammer in flight could support more than just the rigid body mode, then perhaps the hammer-string interaction would give rise to various waveforms on the string as a function of the various relative mode energies, which in turn are presumably a function of the manner in which the key is struck. Thus a more impulsive key strike would introduce more energy in higher hammer vibration modes, and the hammer-string interaction would result in a more complex wave motion in the string and thus a brighter tone. The experimental results of Askenfelt and Jannson, however, indicate that the amplitude of the vibration modes of the modern hammer are so small as to be immeasurable and in any case have no effect on the tone. Interestingly, vibrations were in fact detectable on the more slender hammer shank of the historical forte-piano hammer.

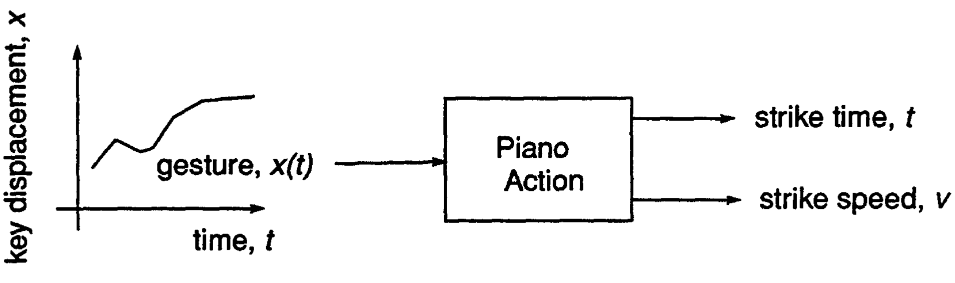


Figure 2.1: *Mapping from Gesture to Sound Parameters as Carried Out by the Piano Action*

certain instant, into a final hammer angular speed occuring at another certain instant. The keypress requires a graph for its full description. It is a function defined over a time interval. But rather than attempting to define it with either a force or a motion history, I shall simply refer to the keypress as a *gesture* since the human finger acts neither as a perfect velocity source nor a perfect force source. Figure 2.1 illustrates the mapping from an input gesture to two output scalars, the hammer strike time and hammer strike speed. Note that the hammer strike time is not directly tied to the initiation time of the gesture; the time interval between initiation and strike is strongly dependent on the shape of the graph of the gesture.

Beyond selecting a pitch by choosing a particular key, the pianist has only to select the two output parameters to completely determine a tone. With the selection of the speed, both the tone intensity and its timbre are jointly determined, for they are coupled, as mentioned above. Although a gesture at the key may be complex and may encode the complex expressive intentions of the performer, its action in the end is completely encapsulated by the two simple scalars to which it gives rise. The mapping from gesture into two scalars is itself complex insofar that it depends on the dynamics of the piano action, a rather daunting (on the surface) dynamical system which will be described in detail and modeled in Chapter 3. For example, similarly scaled gestures will not necessarily give rise to similarly scaled final hammer speeds. But the fact remains that for all practical purposes, the piano action plays a role which is a *severe data reduction.*

Finally, note that this mapping is many to one, that is, there are many ways to strike the key yet produce the same final hammer speed and strike time. The mapping is not, however, one to many. A certain gesture at the key produces a very repeatable final hammer speed and strike time. That the mapping performed by the piano action is repeatable and has as its output an event parameterized by two scalars gives credence to the engineer's claim that the operation of the piano action is 'simple'.

2.1.2 Dominant: The musician's viewpoint

The existence of a complex and highly developed piano technique among pianists, regardless of musical persuasions, seems to contradict the engineer's statement, that all that is involved is the selection of two scalars (strike time and strike speed) for each tone. One must wonder why there exist so many ways to hit the key and, especially, why certain techniques are recommended for their effectiveness at producing certain tone colors. Apparently the behavior of this instrument in the hands of its users is responsive enough to warrant a large amount of attention to skill development. Certainly a large part of that skill lies in selecting the two scalars, hammer strike time and hammer strike speed, but here I am speaking of responsiveness of the piano to the various techniques which are chosen for their coloristic effect, yet intensity invariance. The piano seems to reward its users with tones which vary in color, yet remain constant in intensity according to the use of these chosen techniques. Certain strategies adopted at the keyboard have results which lead the listener to believe that timbre and intensity are being varied independently.

The language which piano teachers speak when trying to aid their students in producing music at the piano is quite obviously built on the assumption that a pianist has independent control over intensity and timbre. Certain techniques are taught or encouraged with the explicit intention of making a passage soft yet dark, or loud yet airy. Piano teachers use such terms despite what they may know of the limitations of the mechanics of the piano action as outlined in the previous subsection.

The existence today of synthesizers and digital pianos allows us to make some further observations in support of the musician's viewpoint. The fact that a digital piano does not seem to reward a pianist's various techniques with independently varying sound parameters suggests that there is indeed something about the piano that has been missed in its representation in digital pianos. I will, however, delay discussions about modern electronic instruments until section 2.2.

2.1.3 Tonic: resolution

How then do we reconcile the positions of the engineer and pianist, if up until now our discussions have given them both support for their disparate claims? The engineer assures us that one scalar (hammer angular speed) cannot independently determine two effects (timbre and intensity). Intensity and timbre are coupled for each tone. The musician, on the other hand, claims to have independent control over intensity and timbre and demonstrates that control, even for the ears of the engineer!

To suggest that both the engineer and musician are both right would seem counterproductive, but that is essentially what I am going to argue here. The musician is merely right (making a truthful statement) about the issue as it occurs and presents itself in another domain: the perceptual domain. The musician is speaking of a perceived phenomenon rather than a physical phenomenon. It is thus with reference to a tenet of cognitive psychology that we resolve the conflict between the engineer and musician, and legitimize the paradox of the piano rather than deny it.

The proposition that psychophysical quantities may have rather remote relationships to the actual physical quantities which underlie them is not a new idea. Cognitive psychologists have defined a distinction between the physical world and the world as it is perceived. The perceived world is established by an individual based on incoming data filtered through the senses. These data are subject to transformations and processing by perceptual operators in the brain or the sensors themselves. The perceived world, however, exists in a manner which is no less legitimate than the real world. It can be operated upon, practiced with, extended, and explored. Furthermore, there exists a tendency by all humans to forget the distinction between the physical and perceived worlds, to accept the perceived world and in fact identify it as the physical world. This tendency to project the perceived world onto the physical world has been called "distal attribution" by [104].

The musician is speaking of this perceived world when he says that a piano tone's intensity and timbre can be independently varied. The fact that the musician is unable to distinguish between perceived reality and physical reality is due to distal attribution. Of interest to us here is the manner in which the transformation from physical parameters to perceived parameters occurs. If we understood this transformation, we could further exploit it in the design of electronic instruments. What parameters of piano music inspire its listeners to identify tones of the same loudness as having different tone color? We would like to identify the physically measurable parameters which the musician (or listener in general) identifies as timbre differences and attributes to certain qualities about key strikes.

I propose that what the musician hears as control over timbre is effected by careful control over timing. For a particular tone, the pianist is indeed able to select only two parameters beyond pitch: the hammer strike speed and the hammer strike time (although the time of tone damping is also important). I propose that by selecting these two parameters, a pianist can produce a percept in his or her listener of timbre being controlled independently of intensity. The contradiction between viewpoints is resolved if we assume that what the listener hears as timbre control is due to fine timing control in the physical domain. The percept depends to large degree on the consideration of not just a single tone, but a group of tones arranged in a musical phrase. By carefully governing the timing overlap of notes as they follow one another, the musician can evoke a certain percept in listeners which will not be labeled in terms of timing at all; it will be labeled by the ear and auditory perceptual centers in terms of the frequency domain, that is, timbre. Although it is not, in physical terms, independent control of timbre and intensity, it is perceived (and labeled) as such.

For example, an arpeggio played with disconnected notes may sound different in timbre than one played with slurred overlapping tones. Tone color differences do indeed exist in the perceived world, despite the fact that they are not supported by the physical phenomena to which they refer.

Finally, we remark that the musician can indeed make valuable contributions to the design of musical instruments, despite the existence of paradoxes between the physics and the perceived music of that instrument. The assumptions which develop in musical pedagogy deserve careful consideration from the engineer interested in instrument building. The languages spoken by the engineer and musician may both be acknowledged as truthful if we allow them to pertain to different domains, and harken to the relationships between those domains with psychophysical studies.

In the present work, I will not contribute directly to the discussion between engineer and musician. I leave these interesting psychophysical studies to future work and to the work of others. However, the premise that control over timing is recognized as control over timbre motivates a deeper consideration of the mapping which the piano action performs on the input gesture. If timing is so critical, it is worth considering issues such as sensitivity of that mapping to slight variations in the input gesture. The gradients of the mapping deserve careful attention in the design of an instrument. Furthermore, the energetic mechanical interactions between pianist and piano become very important to consider. Power exchanges between finger and key are intimately associated with timing. It also becomes plausible that certain techniques are superior for their robustness in the face of disturbances. Disturbances may arise, for example, from the lack of precise control or repeatable control over muscles. This discussion will be revisited in Section 2.5 below.

2.1.4 Literature Review: Paradox of the Piano

Although the topic of the paradox of the piano will not be investigated *per se* in this work, I have collected various sources in the literature which address this topic. The supposition that perceived timbre effects on the piano are due to timing and intensity variations is not new. The ballistic nature of the piano action is quite accessible and has long been understood in qualitative, if not in quantitative terms. I have not found, however, a significant body of literature which addresses the genesis of the timbre variation percepts. This introductory section may be considered a call to arms for this research. It is also hoped that an electronic instrument, as proposed and investigated in this work, could become a tool for such a study.

Many musical acoustics researchers have found the piano paradoxical enough to warrant bringing the issue into the lab for scientific investigation. In 1925, Otto Ortmann, of the Psychological Laboratory of the Peabody Conservatory of Music, published his book Physical Basis of Touch and Tone: an Experime
Ortmann conducted a very thorough investigation which included recordings of the key motion made

with traces left by a vibrating pitch-fork in smoked glass attached to a key. Ortman concluded: "The only factor directly influencing, or responsible for, the vibration of the piano string is the speed with which the hammer leaves the escapement." [78].

In 1934, Harry C. Hart, Melville W. Fuller, and Walter S. Lusby, of the Electrical Engineering department of the University of Pennsylvania, encouraged by Professor Charles N. Weyl, and with the cooperation of a well-known concert pianist, Abram Chasins, conducted some experiments on the piano to determine if the tone as played by a mechanical key striker could be distinguished from that of a live pianist. Using human subjects as auditors, and camera recordings of string motions, they determined that indeed, single notes struck by a finger could be duplicated in every way by a mechanical striker [42].

Carl E. Seashore contributed further to the work of Ortman in 1939 [91]. Together with Tiffin, Seashore used an apparatus for recording hammer motions which they dubbed the Iowa Piano Camera. Seashore is to be commended for his keen understanding of the perceptual factors at play.

Sir James Jeans, aware of the work of Hart et al., sparked an uproar in the piano community with his reading of a lecture to the English Music Teachers' Association on January 7, 1939, which included the quote about the umbrella handle which I used to open Chapter 2. Jeans' lecture was reprinted in full on January 8th and 9th in the New York Times, and was followed up with an interesting article in the Times "This Week in Science" by W. Kaempfert on January 15th of 1939 [53]. Kaempfert, in an attempt to calm the apparent commotion and clear up misunderstandings, pointed out that the scientific results pertained only to a single strike: "No time should be wasted in achieving beauty in single notes, though a good deal of time may be profitably spent in acquiring technique as a whole."

In 1950, J. Helmann, a piano pedagogue at the St. Petersburg Conservatory, took it upon himself to directly refute the claims of James Jeans and company in his book The Consciously Controlled Piano Tone [45]. Helmann bases his evidence on informal psychoacoustic experiments with his students as stimulus producers. Helmann documents an extensive technique of hand motions and speed/force control and a theory to back it up. Helmann was not clear, however, on the distinction between perceptual reality and physical reality.

The commercially very successful reproducing pianos, first introduced around 1910, further fueled the public debate over the mystic qualities of human touch. Reproducing pianos were a technical improvement on the player piano. In addition to the hammer strike times, the reproducing piano was able to record the hammer strike speeds with an extra two tracks on the piano roll. Likewise, the reproducing piano was capable of modulating the pneumatic actuation of the keys. This technology successfully competed with sound recording for about a quarter of a century. Many famous artists made recordings with these pianos. Surviving roles have been restored and used with refurbished reproducing pianos to make audio recordings with modern digital recording technology. These recordings are occasionally hailed today as very accurate reproductions original performances. But in fact, these recordings were usually edited to sound better than the originals, sometimes without the artist's involvement.

Today there exists another generation of reproducing pianos, these implemented with electronics and electromagnetic actuators. The Boesendorfer Recording 190 SE and the Yamaha Disklavier are examples. In contrast to the electronic instruments which will be discussed in the upcoming section, all of these instruments contain a piano action, enabling the recording of the hammer motion. Subsequent perfect reproduction is then possible through design, by trial and error, of an appropriate actuated 'gesture' at the key. So long as the strike speed and strike times of the hammer can be encoded and reproduced, the magic of the piano can be laid bare, as is clear from the above discussion. It is very important to note, however, that the original production of these events by a human player is made possible only by the existence of a piano action in the instrument. During reproduction (recording play-back), the actuators need not use the same gestures as the player, only some known (pre-determined) gestures which will produce the same hammer strike time and speed. The mapping from gesture to hammer strike speed and strike time is many-to-one.

2.2 Synthesizer Shortcomings

So called 'digital pianos' exist *en masse* on the market today. They cater to a significant portion of the keyboard-instrument market formerly occupied solely by the acoustic piano. Digital pianos are essentially synthesizers which feature the best available synthetic (usually sampled) piano sounds and a 'weighted' keyboard. Weighted keyboard is a simple keyboard action whose feel at the keys is made to roughly approximate that of the piano with the incorporation of springs, dashpots, or weights. Digital pianos offer various advantages over their acoustic predecessors: portability, sound-programmability, and occasionally, price.

In this author's opinion, digital pianos represent only a first cut at an electronic instrument modeled after the acoustic piano. Certainly the recent advent of affordable digital sound synthesis has made emulation of the acoustic response of the piano possible [3], but the touch-response of even the best weighted keyboards does not yet compare favorably to the touch-response of a grand

[3] Improvements in the computational reproduction of the piano sound are continually being made by numerous designers and computer music researchers. Most notably, a new synthesis technique known as *physical modeling*, with several proponents at Stanford's Center for Computer Research in Music and Acoustics (CCRMA), [92] [100] promises great gains in the fidelity of audio emulation. The effects of string coupling through the bridge and soundboard and the effects of the nonlinear properties of the hammer felt on the sound can be elegantly integrated into such sound synthesis algorithms.

piano. A second shortcoming of digital pianos, intertwined in origin with the unsatisfactory touch-response, is their implementation of an overly simplistic mapping from gesture to produced sound. Both of these deficiencies factor into why these products cannot answer to the desires of classically (or otherwise highly) trained pianists. In particular, the limited success at emulation exhibited by digital pianos to date motivates careful attention into the dynamical behavior of the target system (the grand piano action) both in terms of its touch-response and the mapping which it performs from gesture to sound parameters. The work of this thesis is primarily intended to remedy the shortcomings of digital pianos in the area of touch-response, yet, as a byproduct of our approach, the improper mapping from gesture to sound parameters will also be amended. Section 2.2.2 will address in detail the mapping from gesture to sound as implemented on modern synthesizers.

In the following subsections, I first state a working definition of *musical instrument* which covers both acoustic and electronic instruments. I then address the importance of the mapping from gesture to sound and set that mapping in context as one of four mappings which every musical instrument performs. Following that, I will discuss in detail a second of the four mappings by making a further observation about digital pianos. This second mapping will become the focus of section 2.3.

2.2.1 Definition of a musical instrument

A rather broad but nonetheless useful definition of a musical instrument is: a device which transforms mechanical energy (especially that gathered from a human operator) into acoustical energy. A musical instrument thus has two interaction ports; first, a mechanical contact, where transfer of mechanical energy from the human operator takes place, and second, an interface between its resonating body and air, where acoustical energy is transferred into the air. This definition underlines the universality of mechanical input of musical instruments (instruments are generally not voice-controlled or remote-controlled, though electronic technology is providing us with many exceptions). This definition also highlights the fact that the design of musical instruments is in large part a study in impedance matching, both at the mechanical input and at the acoustical output. The piano fits easily into this definition, with the point of contact between finger and key as the human/instrument interaction port and the soundboard as the instrument/air interaction port. The voice does not fit so easily into this definition since the operator and sound production equipment are so keenly integrated. But even in the case of the voice, the drawing of boundaries between instrument and operator can be useful for analytical purposes, though these analyses will not be undertaken here.

Considered from the standpoint of conjugate mechanical variables of force and velocity, and conjugate acoustical variables of pressure and flow, a musical instrument really has two inputs and two outputs. Each port is an input/output pair. A mechanical contact (and an acoustical interface) supports both an input and an output simultaneously. Thus, instruments are much more than devices which transform mechanical input into acoustical output.

A musical instrument, then, is an example of a two-port system as it is known in the field of Network Theory. As network theory stipulates, the signs of both conjugate variables can either be positive or negative, but the designation of one as input requires the other variable at that port to be designated as output. This physical requirement is known as the *principle of causality* in the Bond Graph literature [55]. Thus, at the mechanical port, if force is input, then velocity is output and if velocity is input then force is output. The product of the signs of the port variables indicate the power flow direction. Energy can be exchanged (or power can flow) in either direction, but of course only in one direction at any given time

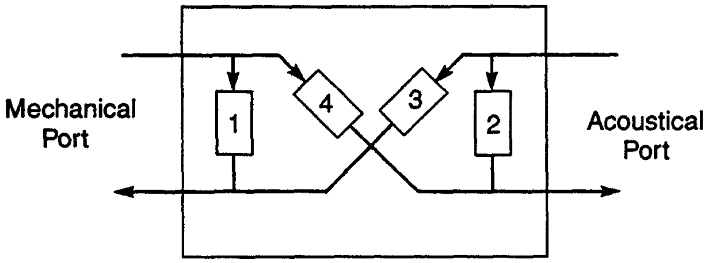
Note that if we consider both conjugate variables at a port to be capable of carrying information (rather than just their product, instantaneous power), then *information* can flow in both directions simultaneously. One of the mechanical output variables potentially supports information flow back from the instrument to the operator by mechanical (haptic) means at the same time that the operator informs the instrument with the other variable. It is not yet clear from psychophysical studies whether humans can independently monitor force and motion at the same time–though it does seem plausible. This question would have to be answered both for long and short term events and recall times.

There are four possible transformations or mappings from input to output. Each will be discussed below. There exists a mapping (or, for linear systems a transfer function) from each of the inputs to each of the outputs. Figure 2.2 shows our two-port definition of a musical instrument with a mechanical and an acoustical port. Note that the two mappings from the acoustical input are not very important for musical instruments–there is very little energy flow in these directions. I have not labeled the inputs as force or velocity, since that is a decision to be made by the analyst/modeler. Of the four mappings, the one most often overlooked is that from mechanical input to mechanical output (note that both input and output are relayed at the single point of mechanical contact.) This mechanical input/output behavior will be our primary concern in this work, but the mapping from mechanical input to *acoustical* output is also a very important part of what makes a piano a piano and a digital piano not a piano.

Synthesizers and digital pianos, by design, are also musical instruments and fall under the above two-port system definition. We must, however, be careful about terminology when analyzing synthesizers and digital pianos since these systems are not only mechano-acoustical but mechano-electro-acoustical. Analogous to the distinction usually made between the physics of the resonating body and the mechanics of the excitation of that body in the study of musical instrument acoustics, depressed. This average velocity is obviously a very crude (in fact, zeroth order) approximation of an input gesture. Neither the mechanical energy relayed by the gesture nor the force input are encoded by velocity sensitivity, and, most importantly, an encoding of the entire gesture (a trajectory over a time interval) is supplanted by a representation with a single scalar. That scalar encoding of the gesture is then typically used as an index in a lookup table to produce the parameter sent to the synthesizer. The user is given access to this lookup table in some electronic instruments. In such cases, the user can set whether the graph of the lookup table is linear, concave, or convex. The output of this lookup table, the MIDI message, then parameterizes the ensuing tone produced by the synthesizer, much like the hammer strike velocity parameterizes the tone produced by a piano string, as discussed in section 2.1. Note, however, that the encoding of the input gesture into a single scalar, right at the outset, cannot be used to emulate the mapping from gesture input to sound output which the piano carries out. Obviously, such a simple input encoding could not implement the same many-to-one mapping. We see that, unfortunately, the paradox of the piano as viewed by an engineer considering the mapping from hammer motion to sound parameters rather than from key motion to sound parameters has been the inspiration for digital pianos. The reduction to a single parameter has been implemented not by a dynamical system such as the piano action, but through the simplistic paradigm of 'velocity sensitivity', a crude representation of the input gesture. The system dynamics of the piano and the interplays between the three other input/output maps of a musical instrument viewed as a two-port system have been disregarded.

This is a grave omission, since the mapping from gesture input to acoustic output is carefully attended to by the musician-users of digital pianos. In fact, with exploratory experiments, it is quite easy for musicians and non-musicians alike to differentiate the mapping implemented on a digital piano from that of an acoustic piano. For example, one can play gradually slower on a synthesizer keyboard and piano keyboard, comparatively monitoring the loudness of the output tone. At slower and slower key depression rates, a synthesizer sounds softer and softer tones, whereas at one point a piano sounds no more tone (the hammer has insufficient kinetic energy to reach the string; no strike occurs.)

In conclusion to this discussion on the mapping from gesture to sound, I would like to note that the correspondence between electronic and acoustic instruments in terms of the excitation means and the sound production means can become quite blurred when considering certain electronic instrument implementations. I suggest that careful scrutiny on the part of electronic instrument designers of each of the input/output pairs of their instruments and the acoustic instruments after which they are modeled will lead to better, more musically expressive designs. Acoustic instruments have one very real advantage over synthetic instruments in that they are implemented by physical means, and thus are subject to some very real restrictions with regard to the available mappings from gesture



1- mapping from mechanical input to mechanical output
2- mapping from acoustical input to acoustical output
3- mapping from acoustical input to mechanical output
4- mapping from mechanical input to acousitcal output

Figure 2.2: *Musical Instrument As Two-Port System*

there is a distinction to be made between 'synthesizer' and 'controller' in the design and use of electronic musical instruments. A 'synthesizer' is a sound synthesis engine; it takes on the function of the resonating body. A 'controller' is a musical interface device (examples of which include a keyboard outfitted with switches, a guitar body with sensored strings, or a clarinet-like tube with pressure transducers) whose purpose is to pick up gestures from the musician. The controller also reduces the gestures of the musician to simple musical parameters which can be relayed to a sound synthesis engine (synthesizer). These parameters are usually encoded as MIDI [4] messages for 'note on', 'note off', 'pitch bend', and so on. Finally, note that the word 'synthesizer' in common usage often denotes both synthesis engine *and* controller, since the synthesis engine is often packaged into a keyboard controller, as with digital pianos.

2.2.2 Mapping from gesture to sound parameters as performed by synthesizers

Modern digital keyboard controllers typically implement a very simple encoding of the mechanical input (gesture) and likewise a simple mapping of that encoded gesture to the MIDI parameters relayed to a synthesizer. The encoding of input gesture commonly implemented is called 'velocity sensitivity'. Two switches are placed at successive positions in the travel of a key so that the time between trip of the two switches gives an estimate of the average velocity with which the key was

[4] MIDI: Musical Instrument Digital Interface, one of the earliest and most successful communications protocols; it implements standardized communication between synthesizer products of various manufacturers.

piano manufacturers are discovering from their outspoken customers just how important the feel of a keyboard really is. K. Yamamoto of Roland, Inc. relates that the design improvement most often requested by customers in recent times is in the area of touch response [108]. High-end synthesizer manufacturers, such as Fanuc of Italy, are even incorporating actual piano actions into their keyboards in order to make them feel right.

Despite the application of assiduous mechanical design, a classically trained pianist will quickly notice that the touch response of a digital piano differs markedly from that of a typical acoustic piano. Thus far, the designs of digital pianos have striven to approximate the feel of the grand piano primarily with the incorporation of passive mechanical components in their keyboard actions. The challenge of design with passive components lies in configuring a spring/damper/mass system which is manufacturable, yet realizes similar dynamics to the piano action. The disparity (as measured either in perceptual or physical terms) however, between the mechanical impedance of these digital piano actions and the acoustic piano action is still quite large. The return force due to the action of gravity on the hammer and key (or static imbalance, as it is known to piano technicians) is comparable, but other factors, such as the balance of spring, damping and inertial component forces, are not consistently emulated. The feel of escapement and hammer bounce (further discussed in Chapter 3), for example, are usually not implemented in digital keyboards.

2.3 Dynamics of the grand piano action

In this section, I will briefly introduce the dynamical modeling of the grand piano action. Modeling of the action is an integral part of our approach to the posed problem of emulating the feel of a grand piano with a synthesizer keyboard. However, a description of our approach to the emulation problem and the role of modeling in that approach will be delayed until section 2.4. I will discuss the reduction of a piano action to a mathematical model (a set of ordinary differential equations) which will be suitable for real-time simulation. This section will describe the various dynamical behaviors of the piano action, identify them as the origins of the 'feel' of the piano, and implicitly highlight the absence of these behaviors and their associated feels in synthesizer keyboards. Even though the reasons for undertaking the construction of dynamical models will not appear until the next section, we undertake the modeling discussion here in particular because of its relevance in the discussion of keyboard touch-response begun in the last section.

to sound. Implementation in real physical hardware instead of electronics and software generally promotes the embodiment of dynamic behavior and interesting interplays between the input/output pairs in an instrument. Musical instruments are, after all, designed to be played by humans who are equipped with very high quality haptic and audio sensors, honed for manipulating and gathering information from real physical systems. Because the excitation is effected through physical means on acoustic instruments (a resonating body, whether string, membrane, plate, structure, or air column, is driven in some manner by another body) it can be argued that humans have greater access to the maps of that instrument through intuition, since intuition is generally built on behaviors of the physical world. By association to the behavior of other mechanical systems with which humans have experience, they may already be familiar with the process of excitation as utilized by a particular acoustic instrument. On the other hand, because electronic instruments draw energy from sources other than the mechanical (control) input (they are usually plugged into the wall,) they can implement more arbitrary mappings from mechanical input to acoustical output. These mappings can be very complex or very simple; they are not constrained to those implementable through mechanical design. The energy contained in the control signal does not have to drive the resonating body. The design of electronic instruments usually involves dealing with more distanced abstractions of gesture encoding and sound production than is the case in the design of acoustic instruments.

To summarize, the words 'impedance matching at the mechanical contact between human and instrument' do not sufficiently cover the consideration needs of the instrument designer since they imply efficient energy exchange as the only goal and do not take into account the cross coupling of internal connections linking input and output. Designers should also talk of 'information exchange capacity' in both directions at the mechanical contact and acoustic interface. Obviously, the information demands of humans are high in both the audio and haptic domains. The degree to which relationships between information appearing on various channels are reflective of physically plausible or organic relationships is also of interest.

In the next section, we will address the mapping in the haptic domain: from mechanical input to mechanical output, but before so doing, we make one more comment about digital keyboards which will lead into the next section.

2.2.3 Judged by feel

Digital pianos are judged not only by their sound, but also by how they feel when played. Musicians become quite preoccupied with the feel of an instrument in their hands, and will readily report on differences they detect and shortcomings which they identify. Indeed, synthesizer and digital

2.3.1 The piano as 2-Port system

I have already defined a musical instrument as a system which interfaces with its environment through two interaction ports: a mechanical port and an acoustical port. Each port has two connections, one for each of two port variables. One connection is an input, the other an output. Internally, four mappings relate each of the outputs to each of the inputs. In the last section, we paid particular attention to the mapping from mechanical input to *acoustical* output as implemented on a synthesizer, highlighting its differences with that mapping as implemented on an acoustic piano. In this section, we will turn our attention to the mapping from mechanical input to *mechanical* output, and shift the focus from synthesizer to acoustic piano.

By illustrating the acoustic piano using the definition of musical instrument as two-port, I will clarify what I have been loosely referring to as the 'feel' or 'touch-response' of the instrument. Remember that the mechanical contact between pianist's finger and piano key is the location for energy transmission between the pianist and piano. Certainly the most central function of the piano is to convert mechanical energy input from the pianist to acoustical energy to be output through the acoustical port (soundboard), but also note that the piano action, in addition to having energy conversion and energy dissipation elements, has numerous energy storage elements. The piano thereby has a capacity to store energy input from the pianist and return it back to the pianist through the contact at the key. Also in the last section, we suggested that both port variables are capable of carrying information, that is, capable of being monitored or tracked by the player. We will therefore center the discussion on the origin of the feel of the piano around the port variables of force and velocity rather than their product, power.

2.3.2 Mechanical Impedance

It is the relationship between the two mechanical port variables, force and velocity, which give rise to the 'feel' of the piano at the key. If we identify the velocity as input and force as output, the feel is characterized conveniently as the mechanical driving point impedance [5].

Mechanical impedance defines the dynamical (history dependent) relationship between force and velocity, with velocity in and force out. Admittance also describes a relationship between force and velocity, except that the force is input and velocity output. In laymen's terms, admittance can be described as the 'give' of a mechanical system. For a given velocity input over time, mechanical

[5] In its strictest sense, the word impedance pertains only to linear systems, expressing the pertinent mass, damping, and spring component forces in response to velocity input. In this work, however, we use the term 'impedance' to refer to the force-velocity relationship of 'nonlinear' systems as well. Nonlinearities may be due to nonlinear constituent laws of spring and damping elements and system discontinuities such as those caused by making and breaking of contact between bodies internal to the piano action.

impedance can be used to express how that system will respond with force. Impedance is also conveniently expressed in the frequency domain and is thus occasionally defined as the frequency dependent mapping from velocity to force. For example, to a certain input motion (velocity), the piano key will respond with a certain interaction force, and do so in a way which is dependent of the history of the input velocity. Note that the definition of mechanical impedance is completely analogous to electrical impedance, where, instead of force and velocity, the port variables are voltage and current.

Vibration Output

Driving point impedance does not fully account for the feel of the piano at the key. The piano is actually a multi-input, multi-output system if we define each key and the soundboard as a port. In addition to the response from the input at that key, there is response from inputs at other keys, and there can be response from energy put into the piano quite some time ago. The piano is, by design, capable of energy storage in the form of vibration of the strings and soundboard at acoustical frequencies; and these vibrations can be felt at the keys. Measurements of typical vibration frequencies and amplitudes at the key of the piano were made by Askenfelt and Jannson and shown to be well above the sensation thresholds for humans [4][6].

Yamaha and other synthesizer manufacturers have attempted to duplicate the vibratory haptic cues of the piano in their digital piano designs by mounting the bass speaker on the keyboard frame in such a way that it produces vibrations at the keys typical of a grand piano.

In this work, however, we will be interested primarily in that portion of the driving point impedance which is definable for a single key; what would be described by the word 'give' rather than 'vibration response'.

2.3.3 The modeling challenge presented by the piano action

The piano action is a compound lever system supporting a myriad of performer/system interaction behaviors. Our goal is to capture in dynamical models those behaviors which are salient for the mechanical driving point impedance at the key. Especially since we will later require these models for real-time simulation, we will construct models which are as simple and computationally efficient as possible, yet preserve the behaviors which we have targeted for emulation. This task will not be

[6] For a particularly interesting discussion of vibration-range haptic response from stringed instruments and its utility, see [4] and other articles in the Spring 1992 issue of Music Perception, devoted fully to Somatosensory Feedback in Musical Performance. In this reference, the use of vibratory cues from the cello, double bass, a singer's chest, and the trumpet are experimentally characterized and compared to human sensation thresholds.

simple since many of the behaviors which the piano action supports are due to effects which are difficult to model with present-day engineering tools.

At first glance, the piano action is like many other mechanical systems in that it stores both kinetic and potential energy, dissipates energy, and transmits energy to other mechanical systems. But there the simplicity ends. Among its component parts are those which must be described by nonlinear constituent equations to capture their effects, most notably the felt and leather couplings. Furthermore, it is a dynamical system which undergoes changing kinematic constraints (bodies make and break contact with one another). It is thus an inherently discontinuous system. In fact, the particular kinematical constraints operative at a given time are not only a function of its configuration, but also a function of the history of the configuration up until that time. For example, depending on how the key is hit, the hammer can be caught on the backcheck or on the repetition lever. The piano action features both holonomic and non-holonomic constraints, especially sliding constraints.

Owing to the non-linear and discontinuous nature of the piano action, it is easy to recognize that inputs and outputs will not scale in linear fashions.

2.3.4 Models of the grand piano action

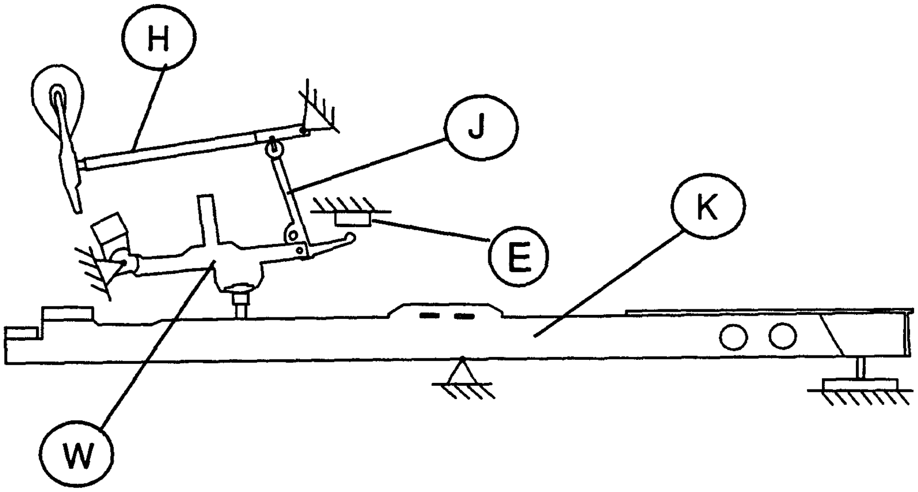


Figure 2.3: *The Elements of the Piano Action*

Figure 2.3 shows a somewhat simplified model of the piano action. It is an assembly of four bodies, known commonly as the key K, whippen W, jack J, and hammer H. The escapement dolly E is fixed to ground. There exist many excellent references covering the history, design, maintenance, and regulation of the piano action. See, for example, [84]. Rather than introducing the behavior of the piano action with reference to Figure 2.3, I will introduce its behavior with a simplified model, one which is rather like a ball bouncing on a paddle. This model will be indicative of those described more fully in Chapter 3.

Beyond the static return force due to the action of gravity on the hammer and key, the feel at the key is dominated by the inertia of the hammer, over which the pianist has a five-times mechanical advantage from the key. The piano action can therefore be modeled, to first approximation, as a static return force along with a simple mass sized to match the effective inertia of the hammer as felt at the key. Because the entire swing of the hammer is less than 25 degrees and that of the key is less than 5 degrees, it is reasonable to approximate this system of rotating levers with a linear system. We assume that all interaction forces and gravity forces act perpendicular to the bodies to which they are applied. We also neglect the effect of sliding between levers.

Figure 2.4 shows a simple linear model for the piano action. The mass m_h is a linear approximation to the rotary inertia of the hammer, scaled by the mechanical advantage squared, or 25. The additional mass m_k represents the inertia of the key, which has a magnitude about 50 percent of the scaled inertia of the hammer. The stiffness || of the connecting spring represents the stiffness of the wooden levers and the felt and leather covers at the various contact points between key, whippen, jack, and hammer. The jack and whippen are not accounted for in this model since their inertia does not contribute significantly to the feel at the key.

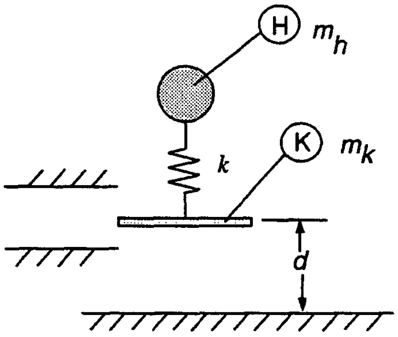


Figure 2.4: *Model 1*

The double-mass model of Figure 2.4, however, is representative of the piano action in only

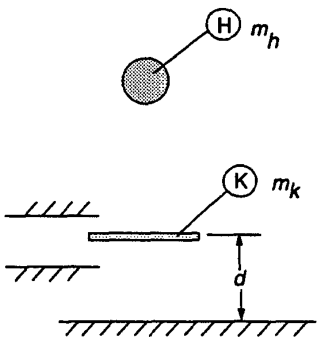


Figure 2.5: *Model 2*

one of its configurations, the initial configuration in which the hammer rests on the jack. This configuration is abandoned at a certain point in the motion of the action —another configuration called 'letoff' takes over. Letoff is the second phase in a sequence of phases or periods of configuration of the action. Letoff begins when the jack meets the escapement dolly E in Figure 2.3 and ends when the hammer flies free of the jack. During letoff, the jack pivots out from under the hammer knuckle where it has been pushing, allowing the third phase to begin, which I shall call 'free-flight'. Here in this introductory section, we will construct a model comprising only two submodels, one for the coupled phase and one for the free-flight phase. Models which include the letoff phase and thereby also capture the effect of letoff in the feel at the key will be presented in Chapter 3.

During free-flight, we represent the piano action by a model in which the hammer is decoupled: a simple mass falling under the action of gravity towards the coupled configuration. Figure 2.5 shows our model for the free-flight phase.

The incorporation of limit stops for the motion K, as seen in Figures 2.4 and 2.5 gives to our model the function of the front and back keybeds (keyrails). The body K is constrained from traveling beyond the upper limit stop, causing H to decouple from K, given that it has enough kinetic energy for lift-off (the spring force shall never be tensile).

This simple 'bouncing ball' model of the piano action supports various interactions between pianist and piano. If the key is pushed down gently, the hammer may never leave the key. If a stronger strike is made, the hammer will leave, and a change in inertia of the key can be detected both when the hammer decouples from the key and later when it lands back onto the key. With a model made up of two submodels we aim to duplicate the effects of the changing kinematic constraint in the piano action arising from the hammer 'bouncing' on the jack. By sequencing back and forth from the coupled to the free-flight submodels as a function of the input, we realize the dynamics of a discontinuous system. For a more detailed description of this two-body model of the piano action, see [33].

Note, however, that the simple model comprising the submodels of Figures 2.4 and 2.5 cannot even be called a piano yet, since it does not have an escapement mechanism. The hammer must be prevented from re-striking the string under the action of a single keypress with an escapement or letoff of some kind. An escapement was the central feature of Bartholomeo Christofori's original invention of the piano in 1705. Other functions of the piano action which will be defined and modeled in Chapter 3 are: letoff and effects of the repetition lever.

2.3.5 Form of the Model

Mechanical system models come in many forms. Because we are interested in running real-time simulations of models such as the one introduced above, we will favor models expressed in forms which lend themselves to efficient simulation. A model can be expressed as an ordinary differential equation (in which the constraint equations are incorporated), a differential algebraic equation (the algebraic constraint equations are adjoined to the differential equations), or a set of coupled second order systems. Each model expression will offer certain advantages for real-time simulation.

Of the various forms in which mechanical system models are expressed, a set of ordinary differential equations (ODEs) in which the constraints are incorporated (used to eliminate dependent coordinates), rather than adjoined, is the simplest. A set of ODEs with incorporated constraints shall be a 'model in independent coordinates', or alternatively, a 'reduced model'. The full model is then formulated as a piece-wise continuous ODE. Discontinuities are allowed at timepoints corresponding to changes in the kinematic constraints. The time periods between the discontinuities are each governed by one of a set of 'submodels', each of these being a continuous ODE constructed to describe the system in one of its constraint conditions.

In the next section, we will address and motivate our approach to real-time simulation of these mechanical models, paying particular attention to the realization of discontinuous systems, *i. e.*, those comprising submodels.

2.4 The Touchback Keyboard

In this section I introduce our solution to the problem of emulating the feel of a grand piano in a synthesizer keyboard which does not include whippens, jacks, and hammers. We do it with motors. Each key is driven by a dedicated motor with which interaction forces are fabricated, under computer control, as a function of the monitored motion of the key. In this manner, a programmable mechanical impedance is created at each key. I will discuss the architecture and algorithms in this section which lead to the emulation of a piano-like mechanical impedance with motors, sensors and computer.

Computer mediated emulation of mechanical impedance is known today as 'haptic display'. This field is quite new; dating back in explicit mention only to about 1980. For a very complete review of haptic interface technology back to its origins, see [71]. Here, I will relay only a few anecdotal notes to place haptic interface technology in perspective.

Robots are generally designed to execute tasks or operate on their environments much like a human and generally in place of a human [26]. To have a human touch or interact with a robot arm is expressly avoided; such activities are (appropriately) regarded as very dangerous. To in fact design a robot, however, for the purpose of interfacing to a human through a mechanical contact is exactly the goal of haptic interface. Haptic interface devices are actually robotic devices– motorized manipulators. But rather than being purposed for manipulation, a haptic interface is purposed to be manipulated by a human and is therefore often called a 'manipulandum'.

Mark Bolas, president of Fakespace Labs, Inc., a firm developing boom-mounted visual displays for application in virtual reality, places the phenomenon of haptic display and virtual reality in an interesting light [12]. Since robots have not been able to enter into our world and take over our chores as was expected back in the '50s, we are now taking it upon ourselves to enter into their world. We are the ones clothing ourselves in their wares: we wear head-mount displays, donn sensored gloves and put on haptic displays in order to interface to the worlds of the computer.

In this section, I will briefly review existing applications of haptic interface technology to music. I will provide a framework description of the basic activities involved in the creation of a touchable virtual object which is modeled after a real-world object. Our own methods will be placed in this framework. Dynamical modeling plays an important role in our methods, and therefore this section provides the motivation for the dynamical modeling of the grand piano discussed in the last section. In the present section, I highlight our formalism and system architecture, especially the manner in which we formulate our models for real-time simulation and incorporate them into the haptic interface hardware and software.

2.4.1 Haptic Interfaces in Music

At Stanford's Center for Computer Research in Music and Acoustics (CCRMA), it was Max Mathew's idea in late 1988 to address the issue of the lacking touch response of synthesizer keyboards with haptic interface technology. Although Professor Mathews was at the time not aware of the then fledgling field of haptic interface, his original sketch, as seen in Figure 2.6, can be recognized as a call for haptic display in music. Professor Mathew's sketch includes a motion sensor, a digital controller, and an actuator in the form of a solenoid.

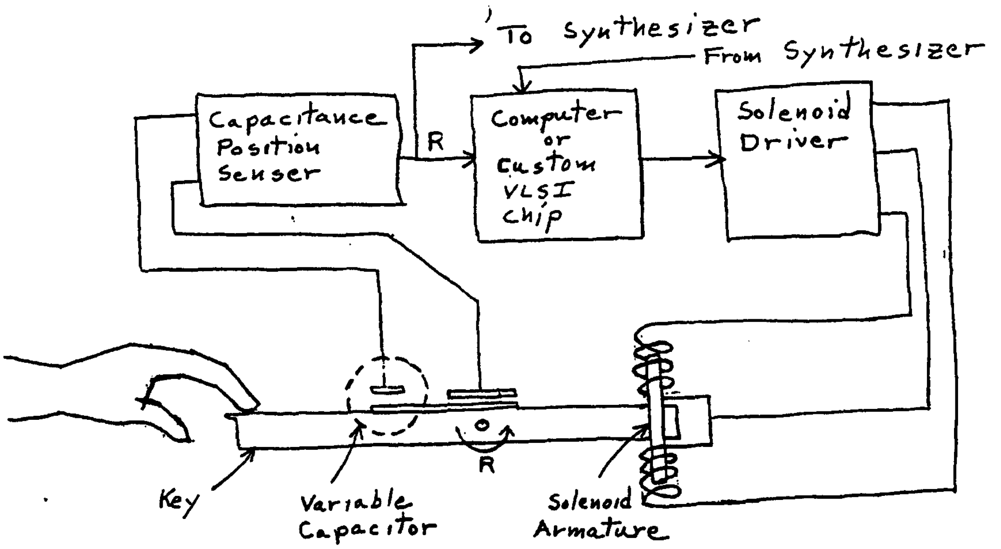


Fig 1 Active Key For Keyboard Musical Instruments

M.V. Mathews
12/28/88

Figure 2.6: *Sketch of Active Key idea by Max Mathews*

Claude Cadoz' group, Association pour la Création et la Rechereche sur les Outiles d'Expression (ACROE) in Grenoble, France began developing haptic interfaces for emulation of virtual musical instruments in 1984 [19]. By 1990, Cadoz' group had constructed a motorized 16-key keyboard of a patented stacked-magnet design and an extensive software support library. The controller software was developed to run on transputers [18]. I will make more comments in Chapter 3 about their simulation algorithms, highlighting our methods and theirs in a common framework. Certainly our work owes much to this early work at ACROE in the area of haptic display for musical application.

Richard Baker holds a US patent for a force-reflecting keyboard [5] which he calls the 'Active Touch Keyboard'. His design includes keys coupled to small motors through a cable and pulley arrangement. An analog controller facilitates the setting of static imbalance and resistance forces. Unfortunately, Baker's design has not been commercialized to date.

Baker's keyboard is not able to emulate the touch-response features of letoff resistance (the rise in reaction force during the letoff phase due to extra friction) or the bounce of the hammer on the jack. The effects of changing kinematic constraints are not captured by the analog controller. We hold features which are localized in the stroke of a key and associated with changing constraints to be central qualities in the touch-response of the piano and primary targets for emulation. Baker's approach using an analog controller, however, is intriguing. An analog controller is not prone to the destabilizing effects of discrete control. By the way, compensation for the destabilizing effects of discrete control occupies a significant portion of this thesis (Chapters 5 and 6). We have used digital control because of its comparatively superior (to date) programmability. Emulation of discontinuous effects (due to changing kinematic constraints) with an analog controller would be possible with nonlinear switching elements such as diodes. An implementation with an analog controller is actually an extension to the work of this thesis.

Alistair Riddell designed and built a 'Meta Action', a set of solenoid-driven hammers to hit strings in an otherwise complete piano. The meta action was designed to take the place of the action and realize manufacturability advantages [85]. We regard the meta action to be an incomplete solution, since it preserves neither the touch-response nor the intricate mapping from gesture to sound parameters of the piano action.

We have designed and built a keyboard-like haptic interface with eight keys which, with its controller, is able to emulate the feel of various keyboard instruments. Interaction with a virtual piano action as is made possible with such a motorized keyboard is a very promising means of re-establishing the kind of relationship between musician and keyboard which supports sententious musical expression. In fact, we believe that haptic display technology will one day become a more viable means of creating the desired touch-response in commercial instruments than mechanical design with passive components. The most notable advantage is the intrinsic programmability of the virtual piano action. The feel of a harpsichord, piano, forte-piano or some altogether new instrument would each be available at the push of a button. But further, the virtual action can be programmed to suit personal preferences. Thus the relationship between touch-response and expressive control can be explored on an individual basis.

2.4.2 System identification, perceptual modeling, and physical modeling

Having posed the basic problem as one of emulating the mechanical impedance of a real-world device with a simple motorized device, we must start with a representation of the impedance of that device, one which will be useful as the core of a controller for a haptic interface. The challenge lies in representing the target device's interaction physics in some form or expression which can be utilized as the controller for the haptic interface. The controller will cause the interface to behave in the user's hand in such a way that the user is led to believe that he is interacting with the target device (when in fact he is just interacting with a motorized manipulandum). By choosing a particular representation, we are in large part also choosing an architecture for the haptic display controller. Likewise, by choosing a method for characterizing the impedance of a device for emulation, we will be favoring certain representations. Since the characterization or modeling process will have such a large influence on the design of a device's implementation as a virtual touchable object, I have identified three distinct methods for characterization.

I propose that the creation of touchable virtual environments encompasses two basic activities: first, a characterization or encapsulation of the target impedance, and second, an incorporation of the characterization data into the software and hardware design of a haptic interface. The first of these activities, characterization, can proceed according to one of three basic paradigms:

1. system identification,
2. perceptual modeling,
3. physical modeling.

Each of these approaches to characterization will later favor a certain design architecture in the haptic display hardware and software.

First a brief note on each method to highlight their differences: The first method involves *system identification* of the target device, and subsequent implementation of the reduced model into the controller of the haptic interface. The second method, *perceptual modeling*, involves assuming a model and then iteratively tuning model parameters for optimal emulation as directed by human subjects making comparisons through haptic exploration. The third method, *physical modeling*, involves building a model of the target system, incorporating that model into a simulator, and running real-time simulations with the human in the loop via the haptic interface.

Let us now consider these characterization methods one at a time.

System Identification

System identification is described by the following procedure. The target system's impedance is characterized empirically, that is, data for the mechanical impedance are collected from the physical device itself into a suitable data structure. That data structure might be a simple lookup table or even a multiply indexed lookup table. The collected data must then be implemented as a formula, algorithm, or lookup table (as befits the chosen data structure) in the controller of the haptic interface. Note that the system identification process can be conveniently accomplished using the haptic interface itself as the probe, if it is instrumented with both force and motion sensors. One can even imagine the ultimate device which would itself feel and characterize the target system, then turn around and present what it felt.

Perceptual Modeling

The second characterization method, *perceptual modeling*, depends on a human subject to make judgements about the quality of whatever attempts are made at emulation. It is with reference to an already rendered virtual object, and its comparative evaluation against a real object or a perceptual (mental) model of such, that a suitable representation for the real object is found, that is, the real object is modeled. This is typically a trial-and-error approach, though measures can sometimes be established for more efficient searches [70]. The use of perceptual models can side-step many difficulties of system identification and physical modeling which depend to a greater degree upon engineering judgment for their success. One may happen upon (or even be directed to by engineering intuition) an algorithm which is found interactively by touch to render a desired effect, such as a rough surface or a particularly hard wall. The selection of the model effectively takes place through evaluation of the rendered effect. An altogether different activity also constitutes perceptual modeling according to this definition: the selection of certain real objects for the purpose of practicing a manipulation task. Surgeons, for example, practice epidural analgesia with a needle through tomatoes and pears, these being their perceptual model for the biological tissues encountered by the needle during that procedure [34].

Physical Modeling

The third characterization method, *physical modeling*, is described by the following procedure. The target dynamical system is modeled in the engineering sense, that is, reduced to a mathematical description such as a set of differential equations. Parameters in the model are estimated from known physical parameters of the system or derived from separate experiments on that system. For example, the moments of inertia and mass properties may either be known or easily determined from the target system with observations of dynamical behavior in response to known inputs. The model is then typically used to formulate an impedance expression (force output is expressed as a function of input motion). Note that a physical model can have many forms, and each form may favor a certain implementation in the haptic interface. Comments on the implementation of a physical model into a haptic interface will be made shortly.

Our Approach

Our approach to the rendering of the impedance of the piano action with haptic display has been inspired by physical modeling, the third characterization method introduced above. We have concentrated on the physical modeling paradigm for the following reasons:

Firstly, we consider the relatively straightforward extension of a model by simple variation of model parameters a significant feature. Also, the manner in which a model grows in complexity is due to an engineer's application of modeling judgement. This implies that accountability is always present in the process, and explosions in complexity are more easily avoided. By the same token, model development is not so easily automated, at least not to the extent possible with either the system identification or perceptual modeling methods. This fact, however, we consider to be a feature rather than a detractor, since at this early stage in the development of haptic interface technology, the incentive for automation is low. Other advantages of physical modeling include the fact that models can be succinctly expressed, shared, and published. Various modeling methods can be applied side by side or by independently working researchers to the same system to confirm results.

The suitability of the particular approach taken depends to a large extent on the nature of the impedance to be rendered. If the impedance is easily expressible as a lookup table, then a system identification approach is straight-forward. For example, multiple lookup tables, one indexed by position, one by velocity, and one by acceleration could be used with their outputs summed to implement a non-linear but superposing impedance. Discontinuous systems (those whose impedance takes drastic jumps as a function of configuration or configuration history), on the other hand, will likely be better rendered with the use of a physical model. Because we felt that the piano action fits into this latter category, the physical modeling approach distinguished itself from the outset of this project.

Finally, very powerful, flexible, and extensible dynamical modeling and analysis tools were available to us early in the undertaking of this project. In particular, the dynamical system analysis program *AUTOLEV* [89] for the generation of equations of motion by automated symbol manipulation presented itself as an extremely useful model construction tool.

2.4.3 Implementation of a physical model

Whereas the implementation of the data structure produced by a system identification activity is relatively straightforward, and a perceptual model depends in any case on a previously cultivated implementation for its production, the implementation of a physical model into a haptic interface is not so simple; it deserves more comment.

Implementation of a physical model takes the form of a real-time simulation of that model. Certain control variables are fed into the simulation in real time, directly from motion sensors on the haptic interface. Examples for the physical modeling approach include flight simulators, although rather than haptic display, flight simulators use motion display.

For each basic form of a physical model, as discussed in section 2.3.5, a corresponding haptic interface implementation can be suggested. If the model is static (and algebraic expression) it may be implemented directly as a control law. If the model is in the form of a differential equation, and if a solution as a function of the input (sensored) variable can be found, it can implemented directly as a control law. If it is a linear differential equation, it can be converted to a discrete formula, and the solution at each time step is obtained by a simple matrix multiplication. If the model is an ordinary differential equation, it can be wrapped with a numerical differential equation solver such as a fourth order Runge-Kutta solver. These implementations will be more thoroughly discussed in Chapter 4. It is also possible to implement either an admittance or and impedance controller for haptic display. These two basic approaches will also be compared and contrasted in Chapter 4.

2.4.4 Our System Capabilities

I will now turn from the discussion of haptic interface implementation in general to a discussion of our implementation in particular.

We obtain the equations of motion governing the behavior of the grand piano action using $AUTOLEV$, paying particular attention to the accommodation of changing kinematic constraints. We have chosen to formulate the equations of motion in their reduced form (incorporating the kinematic constraints) so that they may be integrated by a standard ODE solver. Thus, the rendering of a system with multiple constraint conditions requires the formulation of multiple submodels and a passing of the state information from one submodel to the next at the transition times. Each sub-model governs the behavior only during that time-period which corresponds to the particular kinematic structure for which it was developed. Thus, our simulator is able to handle models described by ODEs which are piece-wise continuous, with the discontinuities occurring at times which are themselves functions of the state. In this manner, a simulation of the piano action is able to account for the changing kinematic constraints that occur when the elements of the action make and break contact with one another.

The rather simplistic model of a bouncing ball developed in the 2.3.4 comprising the two submodels of Figure 2.4 and 2.5 has created a very convincing virtual bouncing object when implemented with a motorized key. Interaction between the ball and user through the key (in this case to be viewed as a paddle handle) includes all the properly timed power exchanges to suggest manipulation of a real ball and paddle. As suggested in the Section 2.3.4, discontinuous (but otherwise linear) bouncing ball model can be thought of as the hammer bouncing on the jack. In summary, we have implemented a unilateral constraint (a gross non-linearity: a contact capable of supporting compressive but not tensile forces) by combining two linear submodels with some management routines for exchanging them in and out of the simulator.

In the most general rigid-body mechanical system, the various constraint conditions may be taken on in any order, depending on how other systems (possibly a user) interact with it. Barzel [9] has addressed the realization of discontinuous systems by simulation of a sequence of ordinary differential equations. In our work, we adopt his nomenclature and combine it with a Finite State Machine (FSM) simulator, which will allow a sequence of conditions or 'states' to be taken in an order which is not known ahead of simulation-time.

2.5 Optimization

We have been working under the premise that the touch-response or feel is a crucial component of the piano, intricately tied to its capacity as a musically expressive instrument. We have been developing means to incorporate the feel of the piano and other keyboard instruments into synthesizer keyboards.

The role of the 'feel' of a keyboard instrument in the music making process is rather subtle, as has been underlined by the discussions in sections 2.1 and 2.2 above. Verily, the mechanical energy exchanges between fingers and keys which give rise to the 'feel' cannot be treated separately from the process of converting from intentions to sound output without losing sight of the investigative purpose. The pianist, who has certain objectives in mind when manipulating the instrument, is subject to the behavioral features or the 'dynamics' of the piano action. The pianist must operate within its constraints, and utilize, to the best of his or her ability, what information about its response that it makes available. We are ultimately interested in the extent to which the pianist has control over the piano. Our measure of control is the degree to which certain objectives are met, especially certain musically significant objectives. Also of interest is the degree to which the pianist can vary the piano's output along chosen lines, that is, lines deemed musically significant.

In particular, we ask: how will the pianist's attempts at varying a single parameter independent of others be met with success, and how are such relationships between mechanical input and audio output dependent upon the relationships between mechanical input and mechanical output of the instrument?

Also note that the control inputs which a pianist uses are the product of a very long development period. Their form is by essence the product of much practice. In fact, there are few human endeavors which enjoy as much devotion and time commitment as piano practice. A professional pianist typically spends three practice hours per day at the keyboard; the professional pianist in training will spend as many as eight hours per day.

2.5.1 Human as Optimizing Controller

Many investigations into human performance have been based on the premise that the human is an optimizing controller. For example, see studies of pilot behavior by McRuer [66]. This leads us naturally to address piano performance with optimal control theory. It seems quite plausible that the purpose of practice is to optimize the piano's response, to bring it as close as possible (according to some measure) to a chosen objective. The pianist strives for optimality despite given constraints and in the face of given detracting influences. In other words, the pianist must work within the bounds posed by performance variations, repeatability, variations between keys on the keyboard or between pianos, and attempt to express a musical interpretation for a particular audience and time.

Indeed, it seems plausible that the mechanical impedance of an object provides the human manipulating that object with a great deal of information as to its behavior and its variety of behavior. It allows the human to develop an internal model or 'intuition' about the system being manipulated. In most every sport, it would be unthinkable to deny the athlete haptic interaction with the sporting equipment. That would be tantamount to robbing the athlete of the source of satisfaction of the game.

Optimal control is a well developed field. There exists an extensive toolbox to handle all sorts of constraints: nonlinear constraints, even model changes (changing kinematic constraints). Optimal control theory has a long history and many proponents. It owes its roots to the calculus of variations, and found extensive application in flight mechanics during the development of aeronautics technology. Today it enjoys a wide application area including economics, mechanical design and process design.

2.5.2 Applications of Optimization Theory: Motivation for Haptic Interface

We have undertaken an optimal control analysis of the pianist/piano system dynamics primarily to elucidate the role of the feel of the piano. The aim of our optimal control study is to give a solid basis for the claim that the feel of a keyboard instrument is important to preserve in electronic keyboard designs. An approach which considers the pianist as optimizer, subject to certain constraints and system dynamics, lays out what we think are the proper roles for each of the players in the game: piano/pianist/musical objective. A block diagram analysis of the pianist as controller, the piano as plant, with both haptic and audio sensors feeding back signals to the controller, is a good starting point. But, if the questions we are interested in asking have to do with the value of feedback, it is a sensitivity analysis or an optimality analysis which we must undertake.

Our intention in using optimization theory is not to develop an optimal control input or find the performance limits, but rather to make certain points about the performer/instrument relationship. We wish to make the role of the feel of the piano explicit in order to motivate the application of haptic interface to this problem. We are not particularly interested in developing a player piano or, as is often the purpose in optimal control studies, an autopilot. We do not plan to use an optimized path in the control law.

We are stepping back to try and answer the questions raised above by applying tools from dynamic optimization theory. We make the plausible assumption that the human control input, after all that practice, is in fact optimized either for efficiency or its effectiveness at attaining some musically significant objective.

By imposing quadratic objectives, and using simple models of the human finger playing a piano or synthesizer action, the optimized control inputs are found. Not surprisingly, the optimized inputs are very much a function of the dynamics of the action–which suggests that piano technique will not transfer to a synthesizer keyboard, as is observed. Various more musically significant objective functions are being studied at present. Treatments of motor noise or lack of repeatability, for example, are current goals. Future experiments are foreseen in which human subjects are asked to repeatedly play their best pianissimo at a force and velocity sensored keyboard. These inputs could then be compared to the analytic 'robust pianissimo'.

2.6 Summary

Humans are admirably equipped to explore and characterize the mechanical properties or behavior of objects in their environment. Our muscles and articulated limbs allow us to manipulate, and our haptic senses provide us with information about an object's mechanical response to our manipulations. Our goals, however, often go beyond system identification or characterization of the mechanical impedance of an object. We may want to influence an object's behavior. Such is the case when playing a musical instrument. While attempting to exact a desired dynamical behavior (and corresponding sound) from an instrument in our hands, we use not only the sound but also the force/motion response in a feedback sense to modify our manipulation. If the musical event is of a long duration in comparison to human response times (typically greater than 200ms), this information may be used by the musician for real-time feedback control. Otherwise, the response information is used for anticipatory control to modify the manipulation the next time around, as with practice and learning. Examples of musical instrument playing in which haptic information is of relatively obvious and immediate value to the player include 'sultando' or 'ricochet' bowing of a string instrument and use of the repetition feature on a grand piano. When an instrument's mechanical behavior has no correspondence to its acoustic behavior or there is no haptic information available, a very valuable channel of communication from instrument to player is lost. This is the case for most synthesizer-based musical instruments. In order to alleviate this deficiency on keyboard synthesizers, yet preserve and even expand their programmability, we are developing a synthesizer keyboard with haptic display.

In the foregoing chapter, we have studied certain psychophysical phenomena underlying the operation and design of the piano. The paradox of the piano, that players seem to have independent control over timbre and intensity, yet the percussive nature of the instrument does not support such independent control, was legitimated rather than denied by acknowledging the involvement of psychophysical and psychoacoustical factors. The fact that players only have control over two scalar parameters for each note, hammer strike time and hammer strike speed (the sound parameters) was discussed. In support of fine control over these parameters, the importance of features in the mapping from mechanical input to mechanical output at the key (impedance) and the mapping from mechanical input to acoustical output (gesture to sound parameters) was highlighted.

We are interested in exploiting these psychophysical phenomena in the design of future electronic instruments, so we have considered their application to date in synthesizer controllers on the market. The unsatisfactory level of control over sound parameters which synthesizer keyboards make available may be attributed to the fact that their design does not include the features of the mappings which are characteristic of the piano.

The mechanical impedance of the piano is mediated by its physics or 'dynamics', so we have studied these by building simple dynamical models of the grand piano action. These dynamical models can be simulated in real-time in a human-in-the-loop control scheme with display through a haptic interface. Our Touchback Keyboard, a prototype haptic interface specialized for the emulation of the feel of the grand piano action, was briefly introduced. Further studies of the principles of control which a human uses can be undertaken with a keyboard-like haptic interface such as the Touchback Keyboard. The observations made in this chapter about the piano may of course be extended to other instruments.

All musical instruments are manipulated through some kind of motor (muscle) control, usually involving a mechanical contact. Through a physical contact, a musician guides the mechanical behavior (and in turn, the acoustical behavior) of the instrument. Even the most contrived and out-of-real-time of electro-acoustic instruments has a some kind of mechanical interface. It is the effectiveness and appropriateness of this mechanical interface with regard to a musician's desire to express himself which concern us here. The haptic senses, by picking up information about the mechanical behavior of the instrument, can glean information about the acoustic behavior if there exists a relationship between the acoustic and mechanical behavior. In most acoustic instruments there does indeed exist a close correspondence between the touch response and the sound response.

The goals of a musical instrument designer may be succinctly stated using two concepts from modern controls theory: controllability and observability. The goal is to maximize controllability and maximize observability. Maximized controllability in an instrument suggests that the performer has very fine control over the instrument's output (and perhaps numerous ways to exercise that fine control). Maximized observability suggests that the instrument makes available to the performer a maximum amount of information about its behavior. And of course one mode by which the instrument may inform the performer is through haptic stimuli. Maximized controllability and maximized observability go hand in hand to enhance the relationship between performer and instrument.

It is not surprising that humans happen to be well equipped for listening to and playing musical instruments since most instruments were invented by humans to be thusly listened to and controlled. As we invent new instruments, however, especially computer-based instruments, we must carefully analyze the musician/instrument relationship. With the computer, after all, we are re-defining the way in which music is composed and performed. In particular, new computer-based music interfaces must consider the existence of mechanical information exchanges in both directions between performer and instrument.

Chapter 3

Dynamics of the Grand Piano Action

3.1 Introduction

In this chapter, I present a dynamical model of the grand piano action. The intended use of this model is for interactive simulation with haptic display: to re-create, with a motorized keyboard, the ***touch response*** of the piano. Additionally, the model will be used to enable a synthesizer to re-create the ***sound response*** of the piano by facilitating a proper mapping from input gesture to sound parameters (the hammer strike velocity and strike time) —where 'proper' is taken to mean reflective of the behavior of the grand piano action. The employment of the model in an interactive simulator and its role as a ***virtual*** piano action will be detailed in the next chapter. In the present chapter, I concentrate on the development of the model and its expression as an impedance operator, suitable for later use in emulating touch response, and as a mapping from gesture to sound parameters, suitable for use in emulating the sound response.

There exist some significant features in the behavior of the piano action which we are interested in capturing in our dynamical model. These features effectively modulate, as a function of key depression and depression rate, the otherwise configuration independent inertial, dissipative and gravity-balance forces which are felt by a finger pushing on the key. I will highlight three such features:

- The piano action relies on an escapement or 'trip' mechanism for its operation. The jack, which initially propels the hammer toward the string by pushing on the hammer knuckle, is pivoted out from under the hammer knuckle just before hammer/string impact in a process called 'letoff'. The hammer will subsequently rebound off of the string, to be caught by the repetition lever or backcheck rather than the jack. Associated with letoff, but only noticeable at low key depression rates, is a period of increased response force just before the key hits the keybed. This brief rise in reaction force is called 'letoff resistance'.

- After a complete key depression, a repetition mechanism (sometimes called the 'double escapement mechanism') facilitates a reset of the jack under the hammer knuckle when the key is allowed to rise off the keybed by a short distance. The hammer is then ready for a repeat strike, and the letoff resistance will once again be encountered if the key is depressed slowly.
- Using only shallow depressions, the hammer may be bounced on the jack and the response forces from the key will suggest to the player that a bouncing object is being manipulated through the key.

Other features of the piano action behavior which will be of interest for emulation include the dependence of the manner in which the key returns to rest position on the manner in which it was hit and released.

To attempt to capture these myriad behaviors in a model presupposes a model form which can exhibit the effects of making and breaking contact between bodies —for at the heart of an escapement mechanism is the making and breaking of contact between system bodies, or what might be called 'changes in kinematic constraint'. [1] To change the operative constraint is the end goal of an escapement mechanism and furthermore, a change in constraint is usually the means of activating an escapement. The fact that changing kinematic constraints play such a large role in the operation and in the mechanical impedance of the piano has greatly influenced our choices regarding modeling approach and simulator architecture. For example, we have decided against system identification or 'black-box' modeling methods. We expect that methods which produce linear models will perform poorly when asked to describe the behavior of the piano action, owing to the discontinuities inherent in its changing constraints. We have decided instead to pursue a multibody dynamical model, parameterized according to properties which can be drawn from the system components individually, such as mass properties, damping and spring coefficients, and dimensions. We use the entire behavior of the piano action, including the motion of all of the bodies, to direct our model construction, rather than just the manifest mechanical impedance and hammer-strike parameters.

[1] Some authors refer to 'changing kinematic constraint' using the term 'change in topology' [35], [102], others as 'constraint addition-deletion' [44]. Some authors describe systems subject to changing constraints with the term 'intermittent motion' [103]. The term 'imposition and relaxation of constraints' has also been used in this context [28]. The Russian literature uses 'variable structure'. 'Variable connectivity' is also used.

Our model takes on a special form in order to capture the behaviors which arise from, and are embodied in, these changing kinematic constraints. The model is actually composed of a set of submodels, each of which describe the piano action in one of its constraint conditions. Accordingly, our simulator has been specially designed for simulating multibody systems with changing kinematic constraints. The simulation of a complete escapement is accomplished by simulating through a sequence of submodels, passing the final conditions of each submodel on as initial conditions for the next submodel at the transition times. In our simulator, the sequencing of the submodels actually takes place interactively, at run-time. A finite state machine (fully defined below) is employed to sequence through the submodels as a function of driving input from the user. Essentially, the finite state machine manages the transition times and the transitions themselves (/ie which submodel takes over from the current submodel).

Section 3.2 below includes a survey of models of the piano action which have appeared in the literature. To prepare for the presentation of our piano action model, section 3.3 reviews the various forms in which a general dynamical model involving constraints may be expressed, paying particular attention to the advantages of each form as regards computational efficiency and ease of handling changing kinematic constraints. Section 3.4 will discuss and defend the particular form which we have chosen for our model and introduce the finite state machine. Each of the submodel components which will be put into service by that finite state machine will be defined. Finally, section 3.5 will present our model of the grand piano action, one submodel at a time. Sections 3.6 and 3.7 present some simulation and experimental results which have been used to develop and verify this piano action model.

3.2 Literature Review: Piano Action Modeling

The small number of analytical investigations into the kinematics and dynamics of the piano action which have appeared in the literature will be examined in this section. Except for the omission of models which remain behind the doors of piano manufacturers such as Steinway, Baldwin, Yamaha, or action manufacturers such as Renner and others, and for the omission of occasional treatments possibly appearing in the numerous piano action patents, the following may be considered a reasonably comprehensive review of analytical investigations into the dynamics of the piano action.

This section will also review the literature pertaining to finite state machines and will present a broad outline of the major forms in which a dynamical model may be expressed.

3.2.1 Dynamical Models of the Piano Action

Walter Pfeiffer published a set of books which treat various aspects of the grand and upright piano actions. Pfeiffer's interest was in uncovering possible design improvements. His book, Whippen and Hammer, deals primarily with the kinematics of sliding contact between the capstan screw and leather-covered whippen heel with gear theory. [81]. Dynamics of the piano action were considered in [80] and [79] using elementary models and applications of conservation of energy.

Dijksterhuis developed a simplified dynamical model of the piano action in [27] (also quoted in [97] and [98]) which accounted for the inertia of the hammer, whippen, jack, and key with an equivalent mass at the point of force application on the key by considering the mechanical advantage of the key over each of these bodies. The friction and gravity forces at the key were modeled as a constant (configuration independent) force. Dijksterhuis reported an equivalent mass of the key, jack, whippen and hammer of 208 grams at the point of application of a playing force on the key. Modulation of the reaction force by changing kinematic constraints was not considered.

Topper and Wills presented a simple model of the piano action in [93]. The action was modeled as a linearized double mass, wherein the key and hammer were modeled as masses coupled by a spring. Model parameters were estimated by experiment and also varied to 'calibrate' the simulated behavior to experimental behavior. Once again, changing kinematic constraints were not considered.

Guido Van den Berghe presented a detailed model of the piano action in [98]. The complete model description can be found in [97]. Van den Berghe's aims were to account for the mapping from gesture to sound parameters with a model which can be simulated in real-time in a synthesizer. As discussed in Chapter 2 of this thesis, and even more thoroughly by Van den Berghe, an implementation of the proper mapping would amount to a vast improvement over the standard 'velocity sensitivity' of today's synthesizers. Van den Berghe used Bond Graph techniques [55] and the mechanical system simulation package DYNAST. Van den Berghe's model does indeed account for changing kinematic constraints. However, his model was not particularly computationally efficient because it belongs to the class of 'coupled force balance' models (defined below). Basically, springs and/or dampers are placed between each massive body, and the Newton-Euler equations are applied for each mass-center, resulting in a model with many degrees of freedom. Van den Berghe reports a 4 hour simulation time on a 486DX33 PC for 1 second of real time. A reduced (linearized) model runs 15 seconds on a DEC 5000/33 for 1 second of real time.

3.2.2 Applications of the Finite State Machine

Finite state machines have a long history of application in design and control. After all, the digital computer is itself a finite state machine, each of its states being determined by the previous state and sensed input. Finite state machines have been used to control manufacturing systems and design fault tolerant event detection and response systems for many years. Recently, the finite state machine has found application in robot control. Schneider [90] applied state table programming techniques to draft out the behavior of a robot in response to events which were detected during that behavior. Thus the finite state machine was used to manage real-time interactions with the robot's environment. The finite state machine also provided a convenient way of integrating high-level user commands, given the intuitive stimulus/event model. Roger Brockett has made use of finite state machines in [14] to develop position and general language-based motion control strategies for robots. Finite state machines find many applications in walking and juggling robots. See work by Raibert [83] and Bühler and Koditscheck [16]. More recently, Hyde and Tremblay et al. [49] and Tremblay and Cutkosky [94] have applied state table programming techniques to the control of robotic hands.

The application of a finite state machine to the handling of changing kinematic constraints in a mechanical system simulation has been suggested as an extension to the PODE formalism in Chapter 14 of Barzel's book [9], which is further reviewed below.

[56]

3.3 Model Form Overview

Due to the importance of real-time simulation within our project goals, we desire a model expression which is computationally efficient, yet still captures the intricate behavior of the piano action. As stated earlier, we are interested in modeling the changing kinematic constraints which give function to the piano action and have an important effect on the mechanical impedance of the piano at the keys. We shall build a multibody dynamical model, modeling each of the wooden elements as rigid bodies, and interspersing lumped parameter springs and dampers to account for the spring-wire, felt and leather components. Having chosen to use a rigid-body model, and given that most of the constraints we wish to model are unilateral (may act to prevent interpenetration of bodies, but not to hold the bodies together), changing kinematic constraints must be handled carefully (see [36] for a thorough discussion). The addition of a kinematic constraint may arise when a collision (typically detected by an interference checker in simulation) occurs between body boundaries. A deletion of a kinematic constraint, however, may happen in one of three ways: the constraint breaks because of insufficient closing force (unilateral condition), the constraint breaks due to relative sliding (one body slides off of the edge of another), or the constraint breaks due to the restitution of an impact. The impact restitution problem must be handled upon collision detection since the bodies have been assumed rigid and the time intervals over which the impact forces will act may be very short, giving rise to discontinuous jumps in certain velocities. The integration routine must be stopped upon the detection of a collision, the momentum-impulse equations solved for the subsequent velocities, and the integration routine re-started. This issue is not a consequence of discrete simulation, but an integral part of analysis using rigid body models.

A multibody dynamical model, embodied in the equations of motion, may be expressed in one of several standard forms. Below I will describe three major forms: the coupled force balance formulation, the dependent coordinate formulation, and the independent coordinate formulation. These various forms may be expressed sometimes as ordinary differential equations (ODEs), and sometimes as differential algebraic equations (DAEs), as explained below. A simulator can be based on each form, though each model expression carries its own advantages with regard to ease of handling changing constraints and computational efficiency. Other factors worth weighing include ease of model construction, ease of implementation in a simulator, availability of algorithms with good numerical properties, and so on. It can also be said that each major modeling technique (Newton-Euler, Lagrangian, or Kane's Method), tends to produce a model in a particular form. The present review, however, will concentrate on model form or expression rather than the formulation process. In general, when the predicted behavior is the same, one form may be translated into another, though when the models are complex, translation is an arduous process.

We have chosen to express our model in an independent coordinate formulation. To provide background for our choice in model form and modeling technique, a few comments about each of the major forms are in order. Brief comments will be made for each form as regards computational efficiency and the manner in which changing kinematic constraints are typically handled.

This small review is not intended to be complete. A comparative study of each of the available model formulations and their associated simulation architectures with regard to their advantages for real-time simulation with haptic display is too large a project to be undertaken here. Contributions to this area have been made by researchers in many fields including computer graphics, numerical methods, robotics, and of course dynamics. Furthermore, interest in real-time and interactive simulation is on the rise, spurned by the digital computer's continuing gains in computing power. Texts with overviews on the field of dynamic simulation are available from the dynamics community. See [43] for a review of numerical methods for real-time mechanical system simulation and [32] for a discussion of both model formulation and simulation methods focused on real-time applications. From the computer graphics community, see [9]. Review papers include [25], which addresses some issues in haptic display. The following outline is drawn from the papers cited therein and each of the above texts (especially [32]) and this outline is unique only in that it carries a broader perspective than any single available review.

For reference in the following discussion, Figure 3.1 is presented.

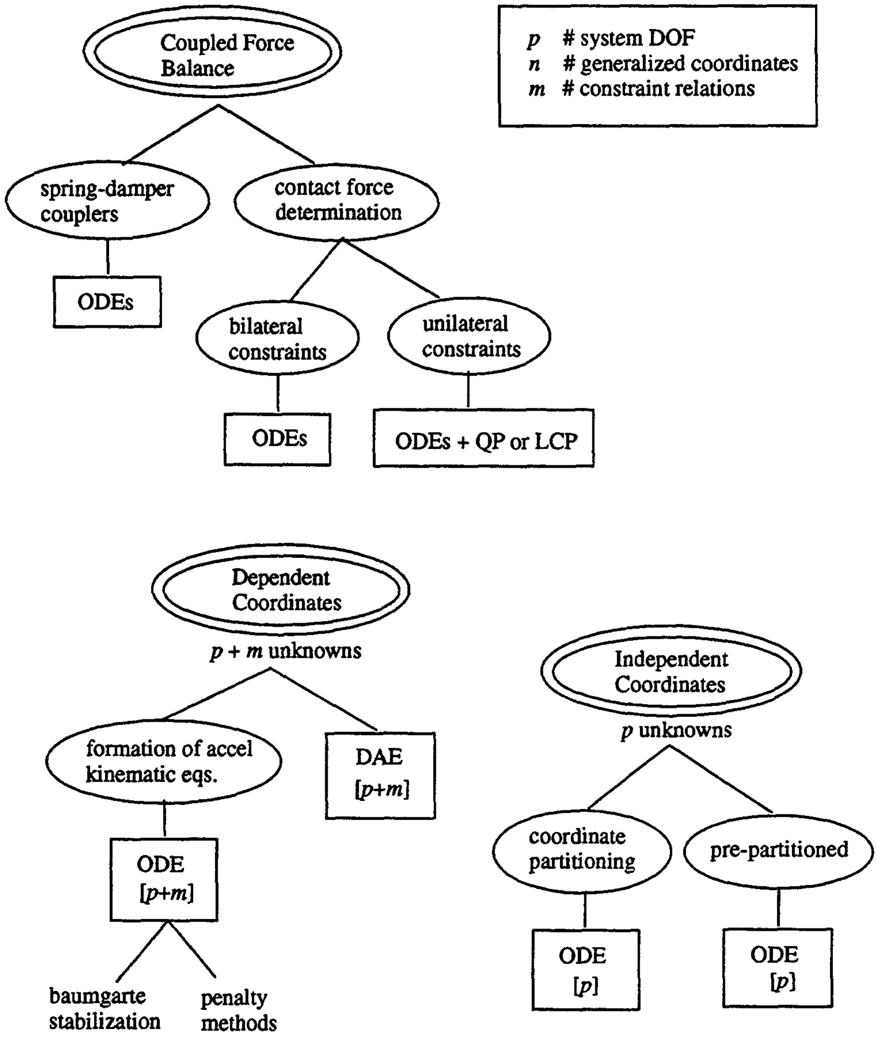


Figure 3.1: *Model Formulations*

3.3.1 Coupled Force Balance Formulations

The term 'Coupled Force Balance' refers to those methods which formulate independent equations of motion for each subsystem. A subsystem is defined as a particle, body, or collection of bodies which within itself is not subject to changing constraints. Changing constraints arise between subsystems [2].

Thus a set of differential equations (models) is produced, one for each subsystem, which are completely independent from one another. Constraint equations relating the motion or configuration of these subsystems are not used to couple subsystem models or formulate system-wide models. Instead, when an interference checker (running in parallel with the dynamic simulation) detects a surface contact between the boundaries of one subsystem and another, an appropriate interaction force is computed in one of two ways (outlined below) and communicated to each of the subsystems for use in their respective (independently running) forward dynamics simulations. The distinguishing factor of a Coupled Force Balance formulation is that, where a constraint condition is subject to change, no constraint relation is employed.

Coupling through Spring-Damper Pairs

The simpler of the two methods for imposing the (possibly changing) 'constraints' in the coupled force balance scheme consists of coupling subsystems through intervening springs or spring-damper pairs when the bodies make contact with one another. In some analyses, these spring-damper pairs are called 'impact pairs' [36] (and references therein). During those times interference between two subsystems is detected, an interaction force is communicated to each of them (for use in their balance) according to the constituent equations of the intervening spring-damper element pair and Newton's third law. To mimic unilateral constraints, these spring-damper pairs are only allowed to exert repulsive forces on the two bodies. When a change from compressive to tensile force is detected, the spring-damper pair is removed. This method obviously produces models with more degrees of freedom than necessary (unless the coupling between bodies really *is* best modeled as compliant or damped), since with the incorporation of such an intervening element, the number of degrees of freedom is not reduced; no constraint equations are imposed. Many examples of this approach may be drawn from the computer graphics community: See Moore and Wilhelms [76]. From the computer music/graphics community, see the work of Cadoz, Luciani, and Florens [20], [63], [30]. Platt and

[2] To derive the equations of motion for each subsystem, various methods are used, including those associated with the dependent coordinate or independent coordinate formulations discussed below (though, as stated above, the 'changing constraints' are not used in the subsystem model formulations). A number of researchers in the computer graphics community, however, consider only particles, or perhaps independent rigid bodies as subsystems. Those researchers which consider only particles apply Newton's second law separately to produce the force balances [20], [63], [30]. When bodies are considered, Newton-Euler principles are used to produce the force balances.

Barr [82] extend these methods to non-rigid body dynamics by applying methods of constrained optimization.

From a rigid-body modeling standpoint, this method may be regarded as approximate since the bodies may in fact interpenetrate slightly. Certain authors therefore call this method non-analytic [6], [7]. Within the computer graphics literature, the term 'penalty method' is generally used to refer to the use of spring-damper pairs to impose (possibly unilateral) constraints. (I find this a somewhat misleading practice, since 'penalty method' is already used to refer to a technique for the stabilization of numerical methods —though certainly the two methods are very closely related).

One advantage of the spring-damper pair is that impulsive forces (forces acting over infinitesimal time intervals) do not arise. The compliant coupler naturally acts to smooth the interaction force between bodies. If the value of the stiffness coefficient of the couplers is increased in an attempt to approximate rigid body behavior, however, the system equations may become 'stiff' and numerically ill-conditioned. A 'stiff' system is one whose dynamical differential equations possess a solution with widely disparate (or widely varying) time constants.

Repeated Impulse

Another method for determining the interaction force between contacting subsystems may be considered at this juncture —it may also be considered 'non-analytic'. This method is not represented in Figure 3.1. Hahn [38] suggests modeling contact forces strictly with impulses. When collisions occur, the time interval for subsystem interaction is assumed to be very short, thus impact forces are involved, giving rise to discontinuities in velocities. (Note that the velocity of a body which is connected, but remotely located to the impact location may change discontinuously). The resulting change in velocities may be found by application of the impulse-momentum equations. A proper treatment must consider the entire system (both subsystems). To handle resting contact (maintained contact), Hahn simply assumes repeated impacts, with high repetition rate. Mirtich and Canny have extended the repeated impulse technique [73] [74] and also combined it with constraint techniques [72].

Contact Force Computation

The second method for imposing the possibly changing constraints between subsystems in the coupled force balance scheme involves computing the interaction forces for subsequent use in the force balances of the contacting subsystems from an inverse dynamics model formulation. This method may be further broken down by whether the constraint condition imposed is bilateral or unilateral.

Bilateral Constraints

By assuming that a bilateral constraint is immediately locked into place when two bodies make contact, an inverse dynamics model may be set up and used to solve for the interaction forces. That is, since the kinematic state is known (relative acceleration between subsystems assumed nil), the interaction forces may be determined. This results in a linear system of equations. The entire system is once again involved in this equation, for each contact point is considered. Singular value decomposition techniques are recommended for robustness near singular configurations [10]. This technique is used by Barzel [10] and Isaacs and Cohen (without treating closed kinematic chains) [51] for application in interactive computer graphics. Both of these authors further exploit the inverse dynamics problem formulation to allow a user-animator to specify desired motion of an object or character. The specified accelerations are used to solve for the forces which would produce that motion. Assembly of subsystems into objects of coupled subsystems may be accomplished by requesting critically damped approach velocities as joints and other 'constraints' are instantiated [10].

Unilateral Constraints

The interaction forces between contacting subsystems may also be determined by solving the inverse dynamics problem while imposing inequality conditions on certain variables. Specifically, non-interpenetration of contacting surfaces and repulsive contact forces (unilateral 'constraints') are stipulated. Denoting the relative normal acceleration between contact points by a_i and the interaction force as f_i, these two conditions can be written:

$$a_i \geq 0, \quad f_i \geq 0 \tag{3.1}$$

where i indexes all contact points. A third condition results from the stipulation that the forces be conservative:

$$f_i a_i = 0 \tag{3.2}$$

In words, this last equation means that, if the interaction force is nonzero, the relative acceleration must be zero (resting contact), else if the force is zero, the acceleration must be positive, in which case the bodies are moving apart. The full system dynamics may be formulated as a linear relation between a vector **a** of contact point accelerations and a vector **f** of interaction forces,

$$a = Af + b \tag{3.3}$$

where A (containing the masses and contact geometries) is symmetric and PSD and b (containing the unknown interaction and inertial forces) is in the column space of A. The above inequality conditions may combined with the system dynamics to formulate a Quadratic Programming (QP) problem (see [6]) or a Linear Complementarity Problem (LCP) (see [7]).

Treatment of the colliding subsystems as rigid bodies necessitates the resolution of impulse forces, using once again the inequalities noted in Equation 3.1. Upon collision, the simulation is stopped and the impulse-momentum relations are used to 'resolve' the impulse forces (determine subsequent system velocities) by solving a QP or LCP. If, after impulse resolution, the subsystems are not accelerating away from one another, they are said to be in resting contact, and the contact forces are found with the solution, once again, of either a QP or LCP.

Lötstedt provides a thorough derivation of the LCP for the impulse resolution and contact force determination problems, also noting that it can be stated as a QP [61]. These unilateral contact force computation methods have been introduced to the computer graphics community by Baraff [6]. Treatments of friction between contacting bodies are presented by Lötstedt in [62] and Baraff in [7]. Lee, Ruspini and Khatib have recently applied the methods of Baraff, citing [6], to robotic simulation in [57].

More comments on the Coupled Force Balance Formulation

With the advent of object-oriented programming techniques, the coupled force balance modeling approach is receiving a fair amount of attention since it fits so naturally into the object-oriented prescript. Most importantly from our viewpoint, changing kinematic constraints are handled quite easily in models expressed as coupled force balances. A communication line between objects (over which the interaction force is relayed to each force balance equation) is simply toggled on and off when interference or clearance is detected. Resolution of impulses may be performed upon detection of a collision. Of course one of the largest challenges in multibody dynamics simulation is the detection of interference and the determination of contact points when two bodies collide (called the collision detection problem), for this determines the points of application of the interaction forces. So long as an effective interference checker is used, the coupled force balance form may easily accommodate points of force application which are not known ahead of run-time, for no constraint equations are formulated.

The number of second order differential equations to be integrated in the coupled force balance scheme equals the number of independent coordinates used. There are no dependent coordinates in this scheme, since direct coupling is effectively eliminated by placing a spring or spring-damper pair between all masses. Accordingly, the number of independent coordinates (degrees of freedom) is large (compared to a formulation involving constraints), and computational efficiency must be regarded as poor.

Noteworthy for this project and its setting in the field of computer music is the fact that the methods of both Cadoz [20], [63], [30] and Van den Berghe [98] fall under the umbrella of coupled force balances, and reportedly make high demands (from our viewpoint) on computational hardware.

3.3.2 Dependent Coordinate Formulations

For the following discussion, it will be necessary to carefully define a few quantities.

The number n of generalized coordinates for a system of bodies S in a reference frame A is the smallest number of scalar quantities such that to every assignment of values to these quantities and the time t there corresponds a definite admissible configuration of S in A (see [54] p. 39). If restrictions are imposed on the positions or orientations which S may occupy, S is said to be subject to configuration constraints, expressed as holonomic constraint equations. When a system S is subject only to configuration constraints, then S is said to be a holonomic system possessing n degrees of freedom in A. Note that for holonomic systems, the number of degrees of freedom, p, is equal to the number of generalized coordinates, n.

If restrictions are imposed on the *motions* of S, then S is said to be subject to motion constraints, expressed as nonholonomic constraint equations. Nonholonomic constraint equations may *also* arise if, in constructing a model, one chooses to use more generalized coordinates than exist degrees of freedom for the model. Then m constraint equations are written to express the m dependent coordinates in terms of the p independent coordinates. The integer m is given by

$$m = n - p \tag{3.4}$$

A model formulation in which the dependent coordinates are treated as unknowns along with the independent coordinates is called a Dependent Coordinate Formulation. Dynamical models in dependent coordinates are often produced using Lagrangian methods. For example, the method of Lagrange multipliers entails *adjoining* the constraints to the n dynamical differential equations resulting in a set of n equations in $(n+m)$ unknowns. Adjoining entails appending the jacobian of the constraint matrix with a pre-multiplying vector of m undetermined coefficients (Lagrange multipliers) to the n Lagrange equations. The m constraint equations themselves may then be used together with the n dynamical equations to bring the number of equations up to the number of unknowns in one of two ways. Firstly, the algebraic constraint equations may be used directly with the differential equations if a Differential Algebraic Equation (DAE) solver is available. DAE solvers are not the most numerically efficient and are not free from stability problems [43]. The second manner in which the constraint equations may be incorporated is by differentiating them twice (in the case of configuration constraints) or once (in the case of motion constraints) to produce m *acceleration* constraint equations which may be integrated along with the dynamical differential equations (because they are now of the same order) in an ODE solver. Due to their derivation through two differentiations (in the case of configuration constraints), the differential constraint equations are unstable, and will require special treatment during integration (see [32], p. 162). Treatments include Baumgarte stabilization [11] and Penalty methods. Baumgarte stabilization essentially attaches the solution via a virtual spring and damper to the manifold of the constraint equations. In the Penalty formulation, the constraint equations are once again incorporated into the dynamical problem directly, penalized by a large factor. Gradient feedback methods to 'constrain' the energy in the simulated system are also available [110]. See Yen, Haug, and Tak [109] for a method to convert DAEs to ODEs on manifolds, which may be used to some advantage.

DAE simulators are favored in the real-time flight simulation and automobile simulation communities, where changing kinematic constraints are occasionally of interest. See, for example [43].

Changing kinematic constraints are handled rather conveniently in the dependent coordinate formulation, because only the adjoined algebraic equations need be swapped out at the transition times. The dynamical differential equations and their state variables continue unaltered. Gilmore and Cipra cover the simulation of planar systems in dependent coordinates with changing kinematic constraints in the two-part paper [35] and [36]. The m constraint equations are automatically (using an 'incidence matrix' containing the continually updated system topology) swapped in and out of the sparse matrix formulation in which the $n+m$ equations have been lined up. The impulse-momentum principle is used to solve for the post impact velocities upon collision detection. Based on the impact response, the post-impact topology is determined (a new constraint may or may not be added).

Haug, Wu, and Yang formulate a model in dependent coordinates in the three-part paper [44], [106] [105] in which changing constraints, impact, and friction are treated in one framework. The separability of the constraint equations from the dynamical equations is used to advantage.

3.3.3 Independent Coordinate Formulations

A model in independent coordinates has only as may dynamical differential equations as there exist degrees of freedom for that model. (Note that some authors call this formulation 'reduced' [103] and [101])

Although it is possible to formulate the Lagrange equations in independent coordinates [103], I will use Kane's equations (which include the notion of generalized speeds) in the following discussion. Kane's method naturally produces models in the independent coordinate formulation.

Generalized speeds are defined by equations of the form

$$u_r = \sum_{s=1}^{p} Y_{rs}\dot{q}_s + Z_r, \quad (r = 1, ...n) \tag{3.5}$$

where Y_{rs} and Z_r are functions of $q_1, ...q_n$ and possibly time t, but not of $u_1, ...u_n$.

The nonholonomic constraint equations active in the system may be used to eliminate the dependent generalized coordinates to produce a formulation in only p independent coordinates, with one proviso: the nonholonomic constraint equations must be either holonomic or 'simple' nonholonomic constraints. A simple nonholonomic constraint is expressible by a relationship between the generalized speeds u_i, $(i = 1, ...n)$ in the following form:

$$u_r = \sum_{s=1}^{p} A_{rs}u_s + B_r, \quad (r = p + 1, ...n) \tag{3.6}$$

where A_{rs} and B_r are functions of $q_1, ...q_n$ and possibly time t, but not of $u_1, ...u_n$.

The dependent coordinates are found during integration either by solving the position problem at each time step, or, more conveniently, by integrating the constraint equations 3.6, which are only first order and stable, along with the differential equations.

Changing kinematic constraints are not so easily handled within the independent coordinate formulation. When the constraint equations change, the entire model must be reformulated; a new set of independent coordinates must be found. Integration of the equations of motion must be stopped, the equations swapped out, and re-started at each change of kinematic constraint.

A method for automatically handling the changing kinematic constraints when the generalized coordinate definitions themselves do not change, but the subset of coordinates which may be considered independent from among the entire set does change, has been proposed by Wehage and Haug [103]. The method is called coordinate partitioning. The jacobian matrix of constraint relations may be solved for the independent rows at each time step, and used to direct and maintain a set of well-conditioned (maximally independent) coordinates.

For handling the impulse-momentum problem in a formulation congruous with Kane's equations (Generalized Impulse, Generalized Momentum), the techniques of Djerassi are available [28].

3.3.4 Closing comments on Overview

This overview suggests that among these broadly categorized model forms, there exists a tradeoff between computational efficiency and ease of accommodating changing constraints. The spring-damper coupler method is simple to implement, may be easily managed for handling changing kinematic constraints without stopping to reformulate model, swap models, or even compute impulses, but it is quite computationally intensive. On the other end of the spectrum are the independent coordinate model formulations which do not easily handle changing constraints, but are maximally computationally efficient.

The computer graphics community has shown the most interest in simulation of changing kinematic constraints, with their interest in animating characters which behave in complex real-world environments. Unfortunately, the model formulation methods of the computer graphics community are generally not very sophisticated. By contrast, attention on changing constraints from traditional dynamicists has been small. There seems to have existed some reluctance to incorporate a constraint into a model if it is subject to change, since its change will of course render the model useless and require re-construction from scratch. Said another way, to incorporate the changing constraint into a model necessitates that such model must be considered transient, and its applicability be continually checked with an inequality condition. With the emergence of computer-aided model formulation, however, model construction need no longer be considered a chore. The checking of inequalities is easily managed by a simulation routine.

An inequality is not so easily handled in the construction of a model, or even in the direct use of a model, as in the inverse dynamics with unilateral conditions as promoted by Baraff [6]. Either quadratic programming or linear complementarity problems arise. To this author, it seems that the inequalities are better handled by the simulation algorithm than by the modeling method or model formulation.

3.4 Our chosen modeling method

We have chosen to construct our piano action model in the independent coordinate formulation, expressed as an ODE, for its computational efficiency and ease of implementation in a simulator, and taken the viewpoint that the difficulty in handling changing kinematic constraints with ODEs is an opportunity to contribute rather than a liablity. Having chosen the independent coordinate formulation and the ODE, we have at our disposal the most standard numerical methods for simulation, leaving room for attention to the relatively complex issue of changing kinematic constraints.

The other influencing factor on our decision of model form was the availability, early on in the project, of an efficient modeling method which naturally produces models in the independent coordinate formulation (Kane's method) and an associated software package for streamlined formulation of the model: AUTOLEV [89]. [3]

[3] I have concentrated my comments above on the model form or expression rather than the methods for construction of the model, but given that the five body piano action is, quite undeniably, a complex system, the choice in modeling technique and associated software package is a very important one. A technique which encourages enough *divide* as

There is another important factor which makes our piano action system ammenable to modeling with independent coordinates, despite its changing kinematic constraints. That is the fact that each of the constraint conditions can be formulated at the outset, allaying the need to run a complex collision detector and constraint formulator during simulation. While the piano action does exhibit changing constraints, it is unlike general simulation of bodies in 3D space (which concerns the computer graphics community) because the number of different possible states is comparatively modest. It is possible to parameterize the point of first contact between bodies with dependent generalized coordinates which are kept up to date during simulation due to the fact that each of the bodies in the piano action are either pivoted to ground or are pivoted to a body which in turn is pivoted to ground. There will be no need to stop and reformulate the equations of motion and constraint equations each time the constraint conditions change. All submodel ODEs (one for each constraint condition) can be formulated ahead of run-time. The set of coordinates will not change, simply the enforcement of the various possible constraint conditions will change.

To account for the changing kinematic constraints with the ODE form, submodels (each a separate ODE) are linked together to form a full ODE which may be said to be only piece-wise continuous. Discontinuities are allowed in both the specification and in the solution (or simulated state trajectory) of the full ODE at the transition times. Our simulator is specially designed to accommodate these piece-wise continuous ODEs. Basically, we wrap a standard ODE solver in an algorithm which can locate events during the solution and manage the exchange of submodels in and out of the solver, keeping the relevant submodel in place and starting each with the proper initial conditions.

Piece-wise continuous ODEs and their use in realizing changing kinematic constraints have been discussed by Barzel in [9], among others. Barzel used the abbreviation PODE for piece-wise continuous ODE. Here I will introduce our extension to Barzel's PODE formalism which I will rather boldly call Event Processing Interactively Sequencing Ordinary Differential Equations (EPISODEs). Whereas the actual order in which the submodels are taken on is pre-determined in the PODE, the order of the sequence is not determined until run-time in the EPISODEs. Only the set of submodel ODEs from which the selections are made in real-time is predetermined in an EPISODE. To decide which, from among the set of all submodels, is to be the next submodel for simulation, (to manage

well as *conquer* is to be preferred. The responsibility for certain modeling decisions should be kept in the hands of the analyst while the drudgery of algebraic manipulation is alleviated. Some of the more powerful modeling software packages available today were not designed in this spirit. It can be argued that some dynamical modeling tools take too much responsibility off the hands of the user-analyist, decreasing net effectiveness.

The Bond Graph methods employed by Ven den Berghe likewise encourage a divide and conquer approach. However, Bond Graphs [55] are rather unwieldy in two dimensions, since they are based on power relations (effort times flow, or force times velocity) rather than vector formulations. Bond Graphs excell in the modeling of systems which include interaction between subsystems of various domains (electrical, pneumatic, mechanical, etc).

the sequencing of submodels) an EPISODE simulator includes a finite state machine.

3.4.1 The Finite State Machine

A finite state machine (FSM) is a system capable of taking on a finite number of states in a dynamically determined (event-driven) sequence of transitions from a particular state to certain others of a set of possible states [4]. A finite state machine is fully specified by its state transition graph, an example of which is shown in Figure 3.2. The finite state model of Figure 3.2 has been drawn to represent the event-driven sequencing of the kinematic constraint conditions in a simplified grand piano action. Only three bodies, H, K, and B, representing the hammer key, and keybed, are considered in this particular FSM. This FSM posesses four states (the four large ovals). The existence of a constraint condition between bodies is noted in Figure 3.2 by a line connecting the body-signifying letters within an oval. The transition paths are denoted with arrows, with a conditional test at the base of each arrow. Starting from a particular state, satisfaction of a conditional belonging to that state will cause the system to take on the state pointed to by the arrow associated with that conditional.

For example, starting in the state in which the key is coupled to the hammer (right oval), either the interaction force between hammer and key f_{KH} will become tensile or the key will hit the keybed (K-B interference) first, depending on how the key is manipulated. From the state in which the hammer is free and the key has not yet hit the keybed (top oval), again: either of the two other states may turn out to be the next state, depending on user interaction at the key.

From this FSM, it is apparent how the constraint conditions may be taken on in various sequences, depending on how other systems (possibly a user) interact with it during run-time. Another advantage of the incorporation of a FSM into a simulator is that it may sometimes be used to reduce the complexity of the submodels themselves. One or more independent generalized coordinates can be dropped from the description of a mechanism if sequencing rules can be deduced from the remaining coordinates. This point will be illustrated with another example taken from musical instrument design: the harpsichord. Figure 3.3 shows the jack of a harpsichord, highlighting each component by name. The operation of the harpsichord jack is quite simple: as the jack is lifted by action of the key, the plectrum meets the string, lifting the string upwards, while bending under the reaction force applied by the string. When the force exceeds a threshold, the string will slip off of the bending plectrum and begin to vibrate. As the jack is once again allowed to lower, the plectrum will not pluck the string, since the spring-loaded tounge and shape of the plectrum will cause the

[4] In the context of the FSM and its use for managing changing kinematic constraints, the word 'state' is taken to mean the operative constraint condition rather than the state vector of generalized coordinates and independent generalized speeds.

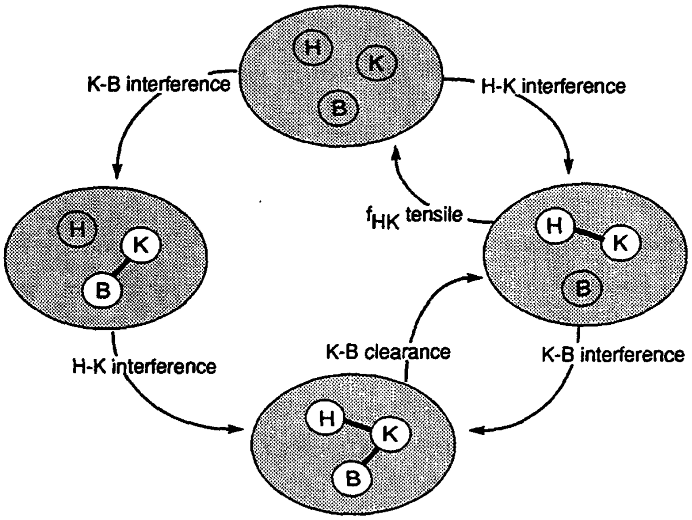


Figure 3.2: *State Transition Graph for a simplified piano action, including a Hammer, Key, and Keybed*

plectrum to easily pivot out of the way upon contacting the string.

To simulate the above mechanism with a dynamical model would entail the modeling of the tongue and the shape of the plectrum, even if the only reaction force one is interested in is the vertical interaction force between key and jack. A simpler approach, incorporating a finite state machine is shown in Figure 3.4. This FSM can be used with a simple static model to create a plucking force on the way up but not on the way down if a pluck has already occured. There is no need to model the tongue.

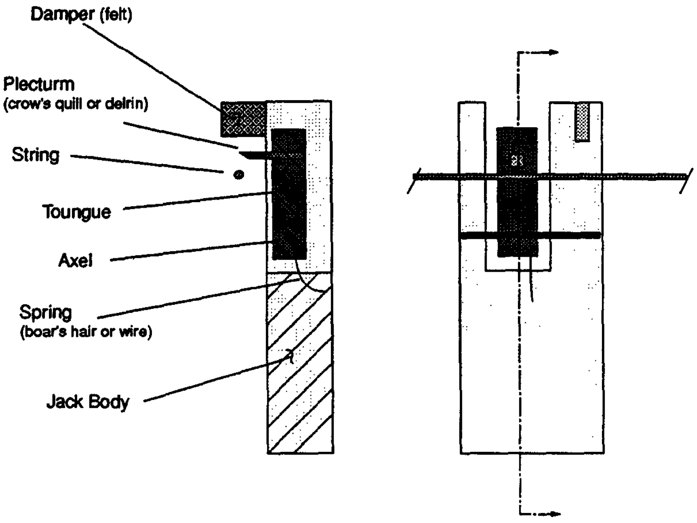


Figure 3.3: *The Harpsichord Jack, shown with plectrum above string (after pluck)*

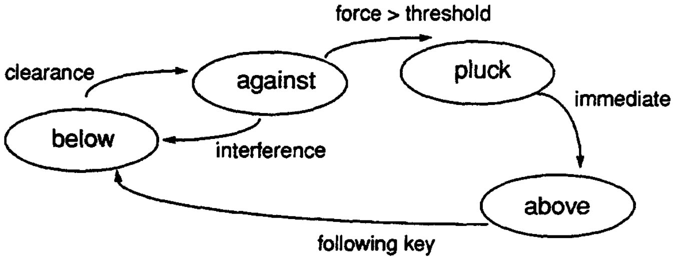


Figure 3.4: *Finite State Machine for the Harpsichord, with state names indicating the position of the plectrum with respect to the string*

3.4.2 EPISODEs

I will now define EPISODEs in detail and outline the construction of a numerical algorithm to solve them. The subscript α (which takes on letters rather than numbers) will be used to enumerate the countable set of submodels $\mathcal{S}$:

$$\mathcal{S} = \{a, b, c...\} \tag{3.7}$$

A complete model is composed of a set of ODEs (submodels), each of which governs the motion in a particular constraint condition.

$$\dot{x}_\alpha = f_\alpha(x_\alpha, u(t)), \quad (\alpha \in \mathcal{S}) \tag{3.8}$$

and a set of readout equations, one for each submodel, which expresses the force output in terms of the state x and input $u(t)$,

$$y_\alpha(t) = r_\alpha(x_\alpha, u(t)), \quad (\alpha \in \mathcal{S}) \tag{3.9}$$

Note that the state x is also subscripted by α since the dimension of x may differ between submodels.

Associated with *each* submodel identified by α is a *set* (indexed by β) of indicator functions which is used to determine the transition times *and* the transition path. The goal of the transition path (a member of $\mathcal{S}$) is denoted by β. Note that the set $\mathcal{T}$ from which β is drawn is a function of α.

$$g_{\alpha\beta}(x_\alpha, u(t)), \quad (\beta \in \mathcal{T}_\alpha), \quad (\mathcal{T}_\alpha \subset \mathcal{S}) \tag{3.10}$$

A *set* of transition functions is used to set up initial conditions for the next submodel from final conditions of the present submodel,

$$h_{\alpha\beta}(x_\alpha) \quad (\beta \in \mathcal{T}_\alpha), \quad (\mathcal{T}_\alpha \subset \mathcal{S}) \tag{3.11}$$

The results of an impulse-momentum solution may be incorporated into the functions h.

The simulator steps forward in time using a numerical ODE solver on the ODE denoted by the present value of α and uses the α readout equation so long as *all* of that ODE's associated indicator functions

$$g_{\alpha\beta}(x_\alpha, u(t)) > 0, \quad (\beta \in \mathcal{T}_\alpha). \tag{3.12}$$

A transition time t_i, $(i = 1, ...)$ signaling the end of the α segment, is found when we detect, for a particular β,

$$g_{\alpha\beta}(x_\alpha, u(t_i)) = 0, \quad (\beta \in \mathcal{T}_\alpha) \tag{3.13}$$

We then switch from the α to the β ODE. The initial conditions for the next ODE are set up as follows:

$$x_\beta(t_i) = x_\alpha(t_i) + h_{\alpha\beta}(x_\alpha(t_i)) \tag{3.14}$$

allowing for addition of state vectors of differing dimension in a straight-forward way.

Of course we cannot find the precise time point at which the event function g_i is identically zero when we are only sampling the indicator function at each integration step. The roots will be crossed over due to the finite step size of the algorithm. If computational time allows, a root finder can be used once a threshold crossing is detected to find t_i to within some prescribed bounds.
Techniques such as these are the subject of Chapters 5 and 6 of this thesis.

3.5 Model Construction

Figure 3.5 shows a profile view of the grand piano action in its rest configuration, highlighting by outline and name each of the elements which will be assumed to be a rigid body in the following analysis. Stick figure representations of each element are also introduced in Figure 3.5. Figure 3.6 shows the same profile view, but highlighting by common name each of the components which will enter our model as a connecting lumped parameter element: springs and dampers. The lumped parameter symbols themselves are also shown in Figure 3.6.

Figure 3.7 shows a schematic representation or stick figure of the piano action, repeated in the same configuration four times. In each subfigure, a different aspect of the model is labeled, but all such aspects are generally applicable. Subfigure a) shows the body and point names, subfigure b) the lumped parameter symbols, subfigure c) the 9 generalized coordinates, and subfigure d) the 27 dimensions. The discussion in each of the following four subsections pertain to a particular subfigure of Figure 3.7.

3.5.1 Generally Applicable Aspects of the Model

Body and Point Names, and Masses

Figure 3.7 a) shows the body names, point names, and any associated masses. The bodies of the action itself are K, W, J, H, and R for the key, whippen, jack, hammer, and repetition lever. E will denote the escapement dolly which is fixed in the newtonian reference frame N. Body M (denoting the manipulandum) is used to drive the model (discussed in further detail below). Because only planar motion shall be considered, a point is sufficient to determine an axis of rotation. Points P_1, P_2, and P_3 are the axes of rotation of bodies K, W, and H, respectively. Point JJ, fixed in W and J, is the axis of rotation for J and point RR, fixed in W, and R, is the axis of rotation for R. Points H^* and K^* are the mass centers of bodies H and K, which carry the masses, M_1,

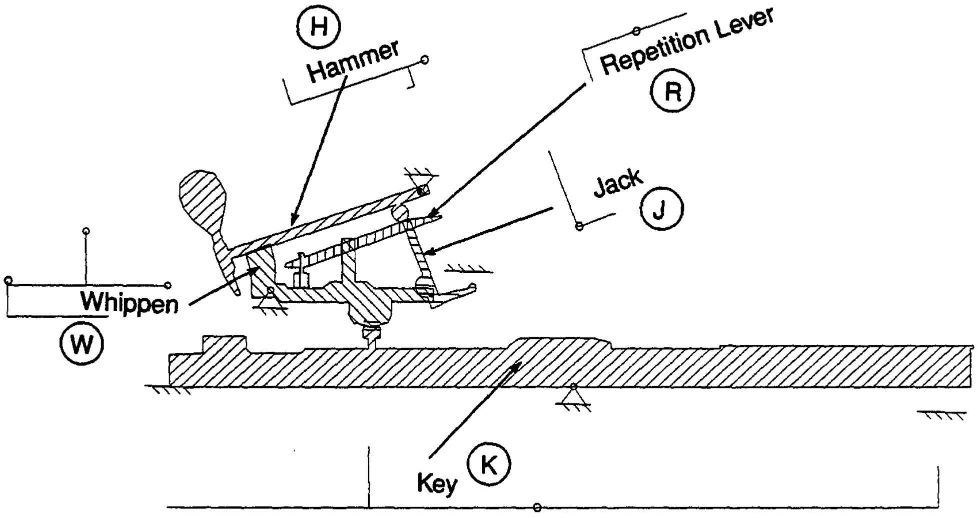


Figure 3.5: *Piano Action Elements: Names, Symbols, and Stick Figures*

and M_2, respectively. Small amounts of mass are assigned to Points RW and JJ to allow equation formulation in 3 degrees of freedom (see Section 3.5.3). Points HJ and JH are the points of contact between H and J, located on H and J, respectively. Similarly, HR and RH are the points of contact (or points of minimum distance, when there is no contact) between H and R, located on H and R, respectively.

Springs and Dampers

Figure 3.7 b) is used to show and define the various springs and dampers which are included in the model. Springs k_2 (keybed back and keybed front), k_6 (repetition lever bed) and k_3 (repetition lever/hammer interface) are special unilateral springs. These unilateral springs are set up with *if* statements in the run-time code. For example, if interference is detected between the key front and the keybed, a spring force proportional to that interference according to coefficient k_2 will act on the key, opposing increased interference. Springs k_2, k_6, and k_7 are typical of the kind of 'constraints' used in the coupled force balance formulation. Indeed, the repetition lever R is implemented in the

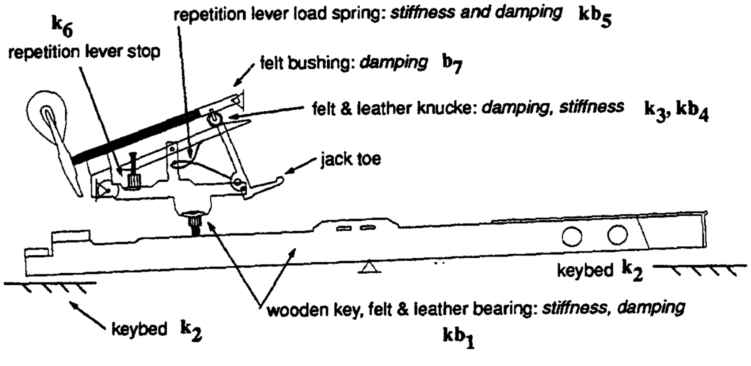

k6
repetition lever stop
repetition lever load spring: *stiffness and damping* kb5
felt bushing: *damping* b7
felt & leather knucke: *damping, stiffness* k3, kb4
jack toe
keybed k2
wooden key, felt & leather bearing: *stiffness, damping*
kb1
keybed k2

Figure 3.6: *Piano Action Components*

manner typical of the coupled force balance approach, increasing the number of system degrees of freedom by one. Had the repetition lever been added using a slider between R and H, no extra degree of freedom would have been required. [5]

Likewise, Body M (the manipulandum), pivoted about point P1, has been added to the model (along with the torsional spring k_1 and torsional damper b_1 which couple M and K) in order to conveniently express the force which will be displayed to the user through the haptic display device. Alternatively, body K could have been driven directly by the user, resulting in a zero degrees of freedom system. In that case, the inverse dynamics problem would be solved to determine the reaction forces to this user-determined input motion. The function of the spring damper pair kb_1

[5] Note that the repetition lever has been added to our model using a 'coupled force balance' approach rather than an 'independent coordinate' approach. The ease of implementation using unilateral springs prompted this decision even though the computational efficiency was thereby somewhat degraded. This compromise turned out to be comfortable given our present computing power. In order to implement the repetition lever using the independent coordinate approach, several more states would need to be added to the finite state machine given the various constraint combinations which could occur.

will be further highlighted in Chapter 4.

Rather than using an auxiliary generalized speed to bring the interaction force between J and H into evidence (for use in an indicator function), spring k_4 has been added to the model. The J-H interaction force is conveniently expressed as the extension this spring, q_4 (see Figure 3.7 c)) times k_4. Notice that the addition of spring k_4 has also increased the number of degrees of freedom by one.

Generalized Coordinates

Figure 3.7 c) shows the generalized coordinates q_1 through q_9 which are used to specify the configuration of the 5 bodies. The angle D which locates body M is the specified input. The radian measures of five angles q_1, q_2, q_3, q_5, and q_6 are used to locate each of K, H, R, W, and J with respect to the horizontal. Displacement q_4 is the extension from rest position of the spring k_4. Displacement q_7 locates a frictionless slider $S1$ which connects K to W. Displacement q_8 locates a frictionless slider $S2$ which connects J to the horizontal line E fixed in N. Displacement q_9 locates a frictionless slider $S3$ which connects J to H. The generalized coordinates will be further discussed below in the text pertaining to Figure 3.8.

Dimensions

Figure 3.7 d) is used to highlight all dimensions used in the model.

3.5.2 Comments particular to each submodel or phase

The motion of the piano action will be broken into four phases. Each phase will cover a certain kinematic constraint condition. The phases will be termed 'acceleration', 'letoff', 'catch', and 'reset'. These phases will also be referred to simply as 'A', 'B', 'C', and 'D'.

Figure 3.8 shows another four stick figures of the piano action, each one drawn in a configuration typical of one of the motion phases. The phases differentiate themselves from one another by the existence or non-existence of a constraint, and thus in Figure 3.8, the subfigures differentiate themselves from one another by the existence or non-existence of a slider.

All of the sliders $S1$, $S2$, and $S3$ are extant or active only during one phase of the motion of the action: during letoff (as shown in Figure 3.8. Despite the fact that less than 9 coordinates are needed during phases A, B, and D, all 9 coordinates are used, in the same order, for all four submodels. In the case where sliders are missing, generalized coordinates tracking their displacement are not needed. Generalized coordinates are nevertheless used and made to track a 'would-be' slider through the use of an 'artificial' constraint. In Figure 3.8, the generalized coordinates associated with slider positions (q_7, q_8, and q_9) are shown in all four subfigures. However, when these generalized coordinates are maintained by an artificial constraint (no slider active), they are shown with dotted lines.

For example, q_8 and q_9 are used to locate points BJ and HJ on E and H, which are closest to the head or toe of J, respectively. Using artificial constraints, the passing of the state vector from one submodel to the next may take place without any transition functions h_α. Table 3.1 shows the three phases of motion and the existence of each of the sliders to further clarify the manner in which the kinematic constraints evolve. The manner in which the generalized coordinates q_8 and q_9 are constrained, whether by kinematic loop equation or by artificial constraint is also noted in Table 3.1.

Table 3.1: **Breakdown of constraint equations by type and submodel**

Motion Phase	DOF	Active Sliders	no. slider constraints	no. artificial constraints	artificially constrained
A acceleration	3	S1, S2	4	2	q_8
B letoff	3	S1, S2, S3	6	0	-
C catch	3	S1, S3	4	2	q_9
D return	3	S1	2	4	q_8, q_9

Acceleration

Figure 3.8 a) shows the action in the *acceleration* phase, with slider $S1$ connecting K to W. Note that during the *acceleration* phase, the relative angle between J and W remains constant. The *acceleration* phase ends when the toe of the jack first makes contact with E.

Letoff

Figure 3.8 b) shows the action in a configuration typical of the *letoff* phase. During *letoff*, J and E remain in contact and move with respect to each other on a line, regulated by a slider $S2$. The letoff phase reigns while there is contact between the jack and escapement dolly, and passes to the catch phase either when the interaction forces between the jack and hammer at the knuckle are no longer compressive, or the jack slips out from under the hammer knuckle, which is detected with a threshold on the magnitude of generalized coordinate q_9.

Catch

Figure 3.8 c) shows the action stick figure in a configuration typical of the *catch* phase. This third phase is characterized by free flight of the hammer and further motion of the key, whippen and jack until the jack toe drops below E as detected by a limit on the angle between J and W.

Reset

Figure 3.8 d) shows the action in a configuration typical of *reset*. During reset, only slider $S1$ is at play. Body H continues to settle on R. The reset phase ends when the distance between points HJ and JH drops below a certain threshold, set low to ensure satisfaction of the corresponding constraint which is subsequently enforced –during the follow-on acceleration phase.

3.5.3 Equation formulation

Generalized speeds u_i $(i = 1, ...9)$ are formed as a function of the generalized coordinates simply by setting

$$u_i = \dot{q}_i \quad (i = 1, ...9) \tag{3.15}$$

Six kinematic constraint equations are used to express u_i $(i = 4, ...9)$ in terms of u_r $(j = 1.., 3)$. Expressions are found for the velocities of each of the massive points or points to which spring or interaction forces are applied. The partial velocities for each of these points are found, and are used together with the gravitational forces acting at K^* and H^* and the applied and interaction forces to form the generalized active force F_r $(j = 1.., 3)$. Expressions for the accelerations of K^*, H^*, RR, and JJ are found and used to form the generalized inertia force Fr^* $(j = 1.., 3)$. Finally, the dynamical equations of motion are formulated:

$$F_r + F_r^* = 0, \tag{3.16}$$

[54]. Appendix A contains the *AUTOLEV* input files for each of the submodels.

We are now in a position to link the four submodels during simulation. The generalized coordinates are lined up, so we may pass the final conditions on as initial conditions to whatever submodel comes up next. The indicator functions are presented in Table 3.5.3. Figure 3.9 shows the state transition graph linking the four submodels. During the catch phase, a strike of the hammer on a virtual string is facilitated by a simple *if* statement in the run-time code. A struck string event is demarcated by the sounding of a tone by a synthesizer hooked into our hardware setup. At the time of contact, the sign of u_2, the generalized speed associated with H, is reversed to effect a perfectly elastic collision between hammer and string.

Collisions between bodies which occur at the transitions between phases are assumed perfectly plastic. No impact analysis is performed. The plastic assumption is reasonable in the case of the piano action since, by design, impacting surfaces are covered with felt or leather to avoid impulses.

Table 3.2: **Indicator Functions**

Indicator Function	Expression	Description
g_{AB}	$\mathbf{p}^{JB-BJ} \cdot \mathbf{N}_3 < 0$	vertical distance between jack toe and E
g_{BA}	$q_6 - q_5 - q_{WJ}$ ¡ 0	J-W angle thresholded
g_{BC}	$q_4 < 0$, or $q_9 - \mathit{threshold} < 0$	J-H interaction force tensile or slider slips off edge of knuckle
g_{CD}	$q_6 - q_5 - q_{WJ}$ ¡ 0	J-W angle thresholded
g_{DA}	$\lvert p^{JH-HJ} \rvert <$ tolerance	points JH-HJ within range

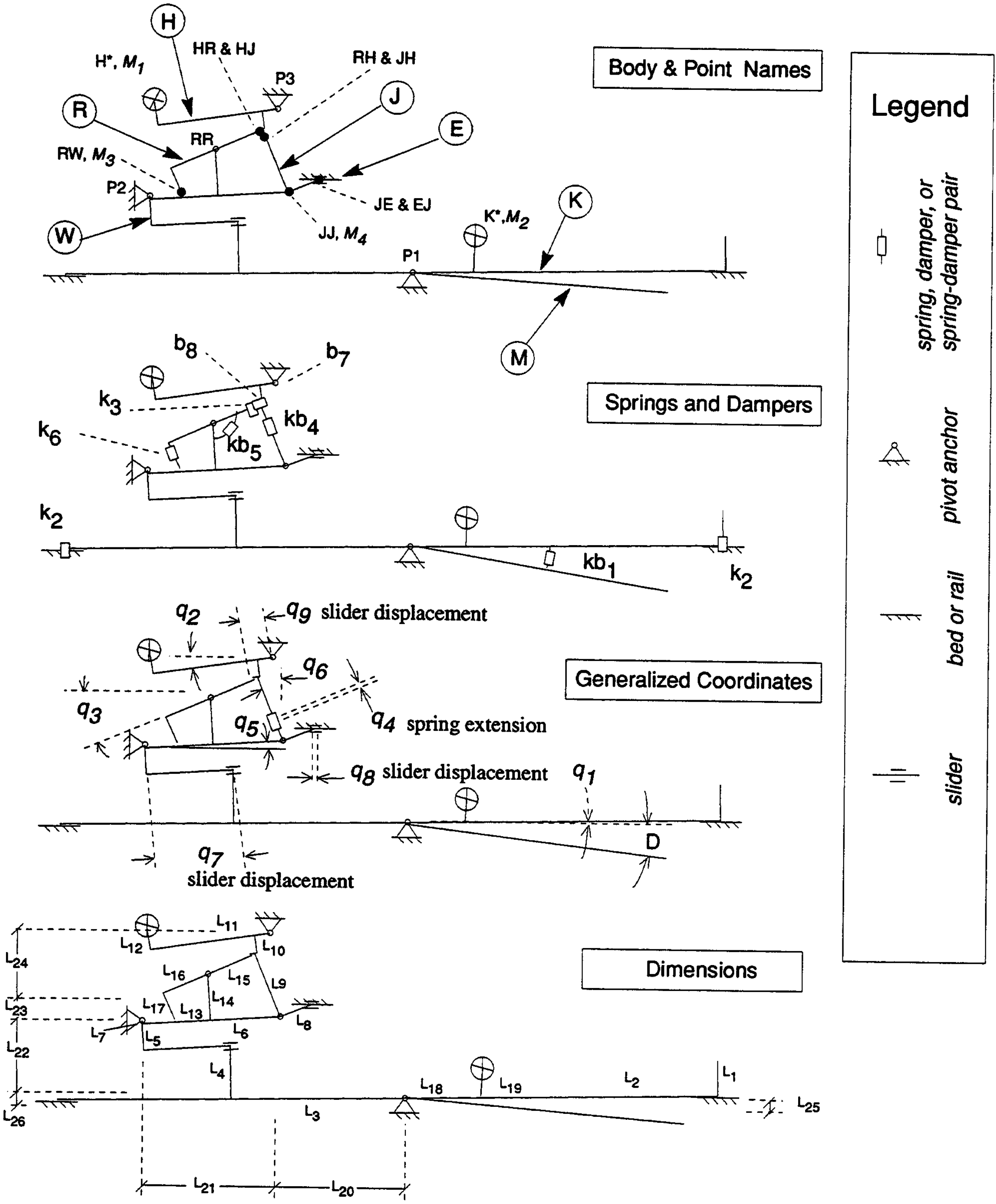


Figure 3.7: *Piano Action Components*

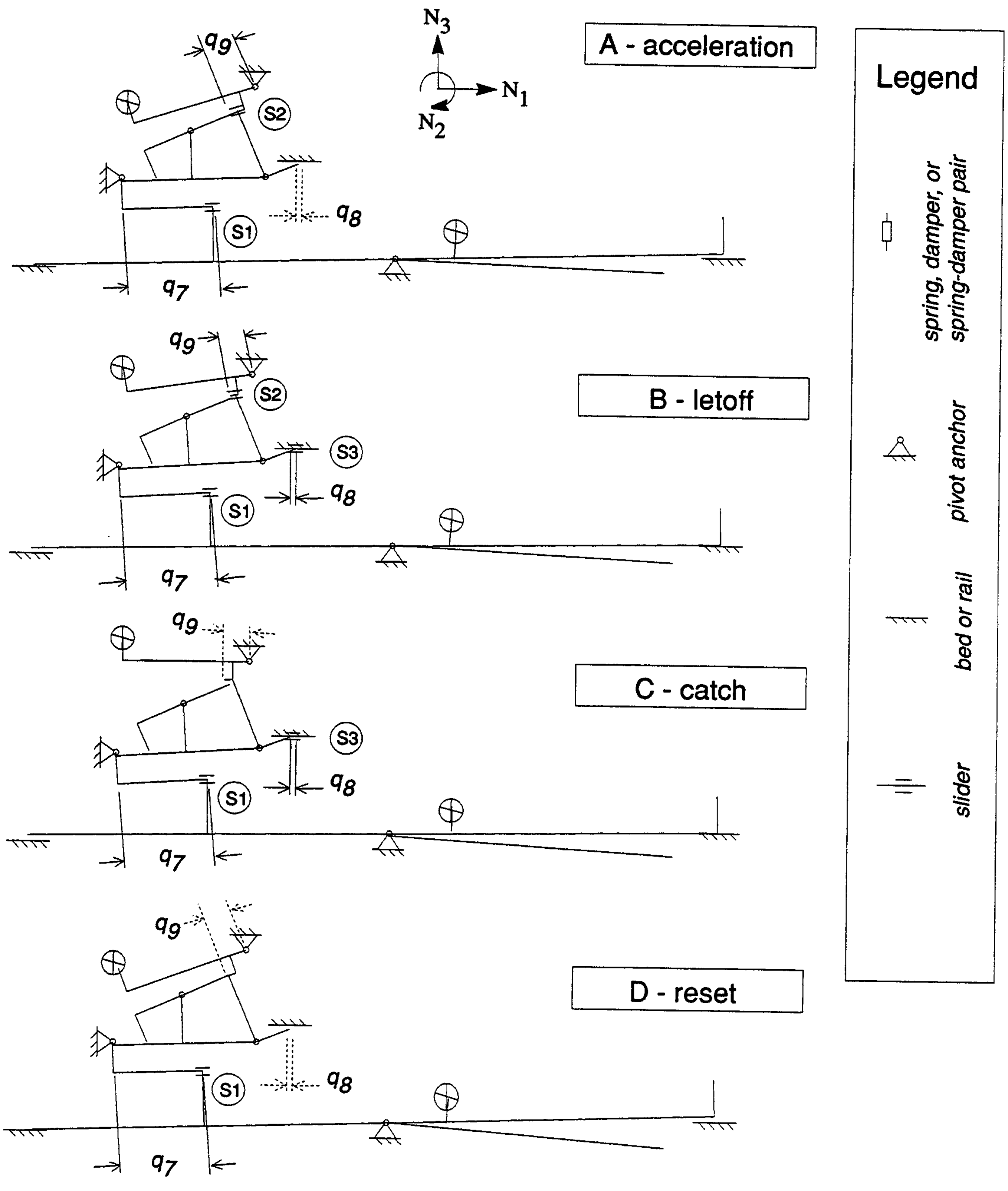


Figure 3.8: *Piano Action Components*

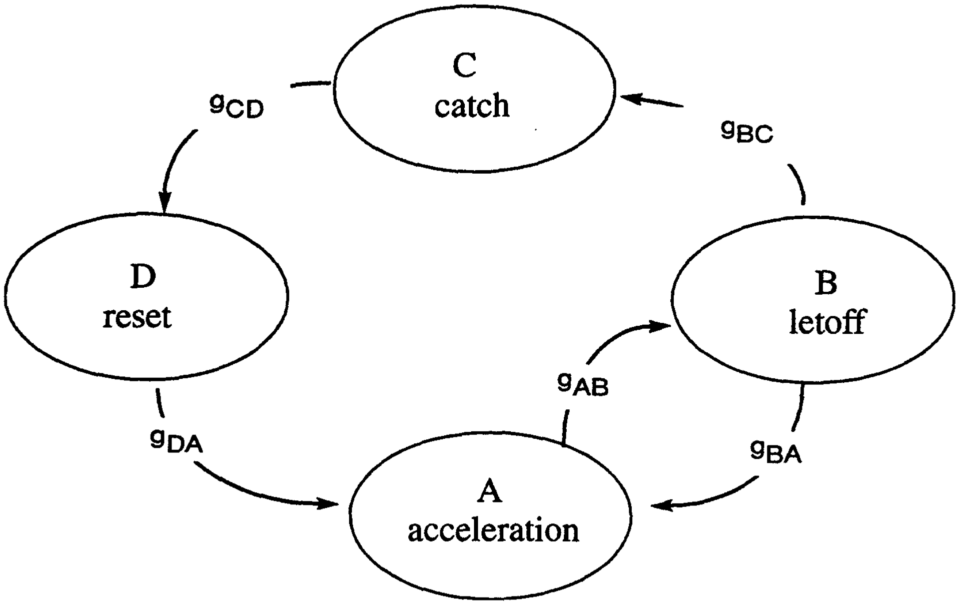


Figure 3.9: *State Transition Graph for the Piano Action*

3.6 Simulator

This section describes the interactive simulator which has been developed for this project. Presently available dynamical system modeling and simulation packages are not designed for real-time simulation, so we built our own simulator from scratch. First, the various components and software tools which are used in the preparation of a new model for simulation are presented. Second, some details of the software architecture are described and finally, some simulation results are presented.

3.6.1 Simulator Components

Figure 3.10 shows the various code components which communicate with the simulator. Basically, construction of a new model for the simulator involves little more than writing the *AUTOLEV* input file, specifying the geometry with AUTOCAD, and invoking various compilers. The simulator then provides for real-time graphical, audio, and haptic interaction with the new model as depicted in Figure 3.11.

Input and Output Files

Paths in Figure 3.10 which terminate on the simulator show the files which are used to add a new model to the simulator's repertoire along with the tools which are used to produce them. *AUTOLEV* 3.0 is used to produce the equations of motion from a command file (model description) as outlined in the previous section. The equations of motion (in the *.EOM file format) are converted into C++ code by a small compiler called *CODEUP* in order to prepare these equations to be compiled with the remaining simulator code [6].

The Model files (*.MDL) contain the geometry to be animated. These files may be generated using AUTOCAD or any other .DXF-format compatible CAD package. Finally, initial conditions and all model parameter values, including dimension, mass, damping, and spring parameter values are loaded in from the (*.DAT) file.

The single stored output from the simulator is a data file shown with an arrow-tail on the simulator block of Figure 3.10. The generalized coordinate, interaction force, indicator function, or virtually any variable trajectory can be stored to disk. Simulation output has been used to place drawings of the bodies into successive frames which have been compiled into Stick-figure animations and full 3-dimensional AUTOCAD animations.

[6] One can imagine future simulator versions which use an interpretive process rather than compilation to incorporate the equations of motion for a new model. For that matter, the task of producing the equations of motion themselves could be taken over by the simulator.

User Interface Devices

The simulator features a host of user interface devices which allow for various kinds of real-time interaction. These are shown linked by double-headed arrow to the simulator in Figure 3.11.

- Sound display is provided through a MIDI-driven synthesizer.
- A graphical user interface facilitates the modification during run-time of virtually all parameter values, even control values such as the time step and display gains. Figure 3.12 shows an example simulator control dialog box. Note that space-efficient access to all parameter values is provided by pull-down buttons.
- An electro-mechanical apparatus employing motors coupled to keys, the design of which will be featured in Chapter 4, provides for haptic interface.
- A scope view provides for real-time graphing of various generalized coordinates or other variables.
- A model view module provides for real-time animation of the geometry loaded from the Model file (*.MDL). We have found real-time visual display to be invaluable for debugging simulated behavior. Synchronous haptic and visual display can provide many clues when something is wrong.

3.6.2 Some Details of the Software Design

The simulator was developed for MicroSoft-Windows using the Borland C++ compiler to run on a Pentium 90 MHz PC. The equations of motion and expressions for the interaction forces are integrated numerically using a Fourth-Order Runge-Kutta algorithm. Care must be taken to ensure that the initial conditions do indeed satisfy the applicable constraint equations. This is accomplished by solving the set of non-linear non-differential constraint equations numerically by the method presented in [54], page 222.

A dialog box chosen from the main menu from several available virtual objects becomes the main simulation controller. The simulation dialog box acquires its functionality by composition rather than by inheritance (See Figure 3.13.) Upon selection of say, the virtual piano action, each of the pertinent submodel objects (differential equation to be integrated) is constructed and composed into a finite state machine. Because the members of the graphical objects, models, and indicator functions are all written as virtual member functions, the simulator object can take advantage of the run-time polymorphism features of C++. Thus, the simulator engine itself is completely generic code, able to operate on any of the models. Once created, a simulator dialog box will be called upon at each iteration of the event loop to execute the following: poll the sensors for current readings, use the current submodel to integrate ahead one time-step, check the indicator function and change submodels if necessary, send a force value per the readout equation to the D/A converters, and finally, perform graphic and sound display functions.

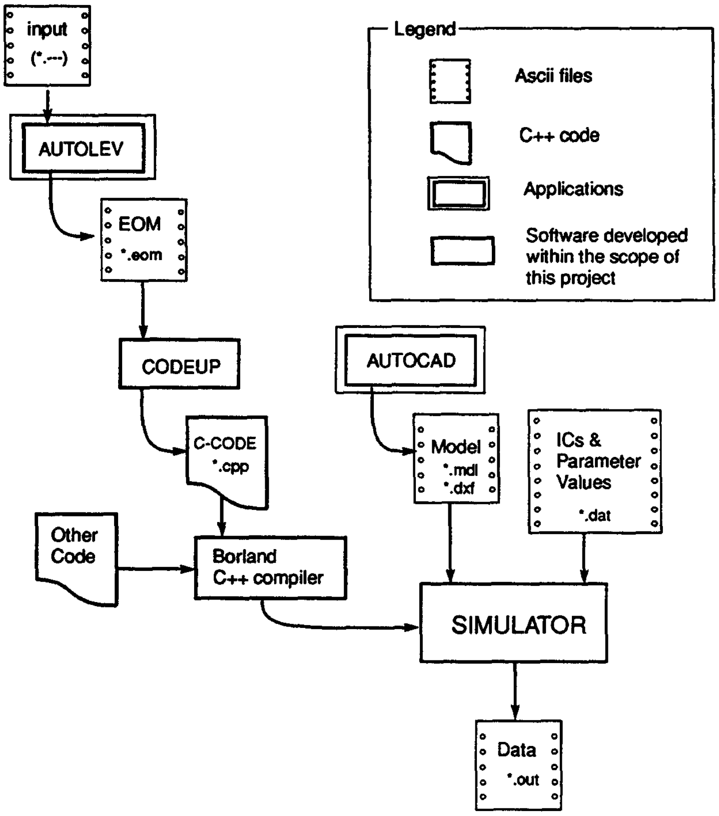


Figure 3.10: *Input and Output Files for the Simulator, and their Producers*

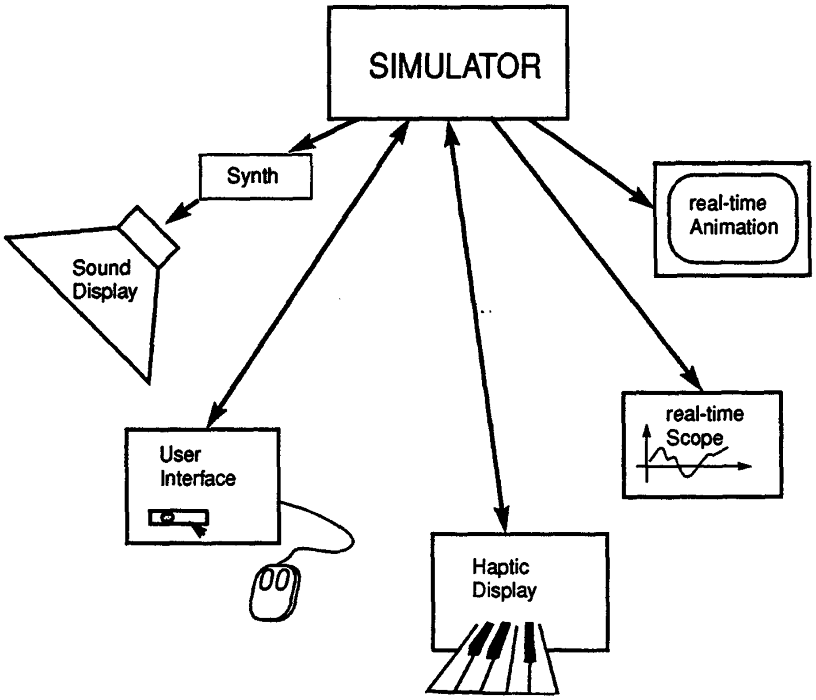
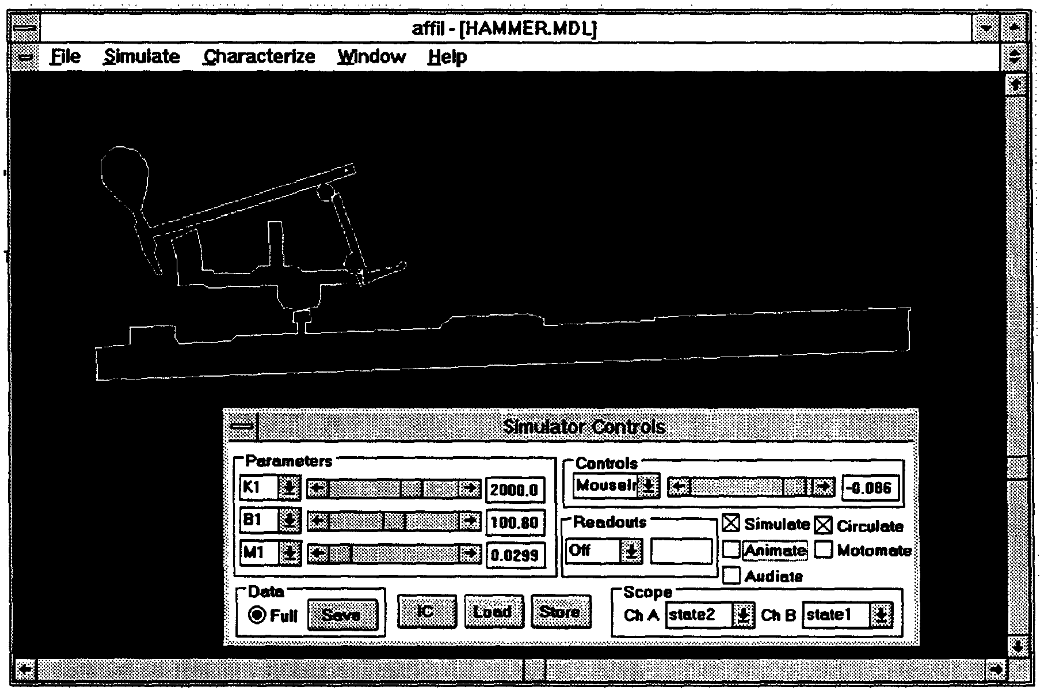


Figure 3.11: *Interface Devices for the Simulator*



Parameters
K1 2000.0
B1 100.80
M1 0.0299

Controls
MouseIr -0.086

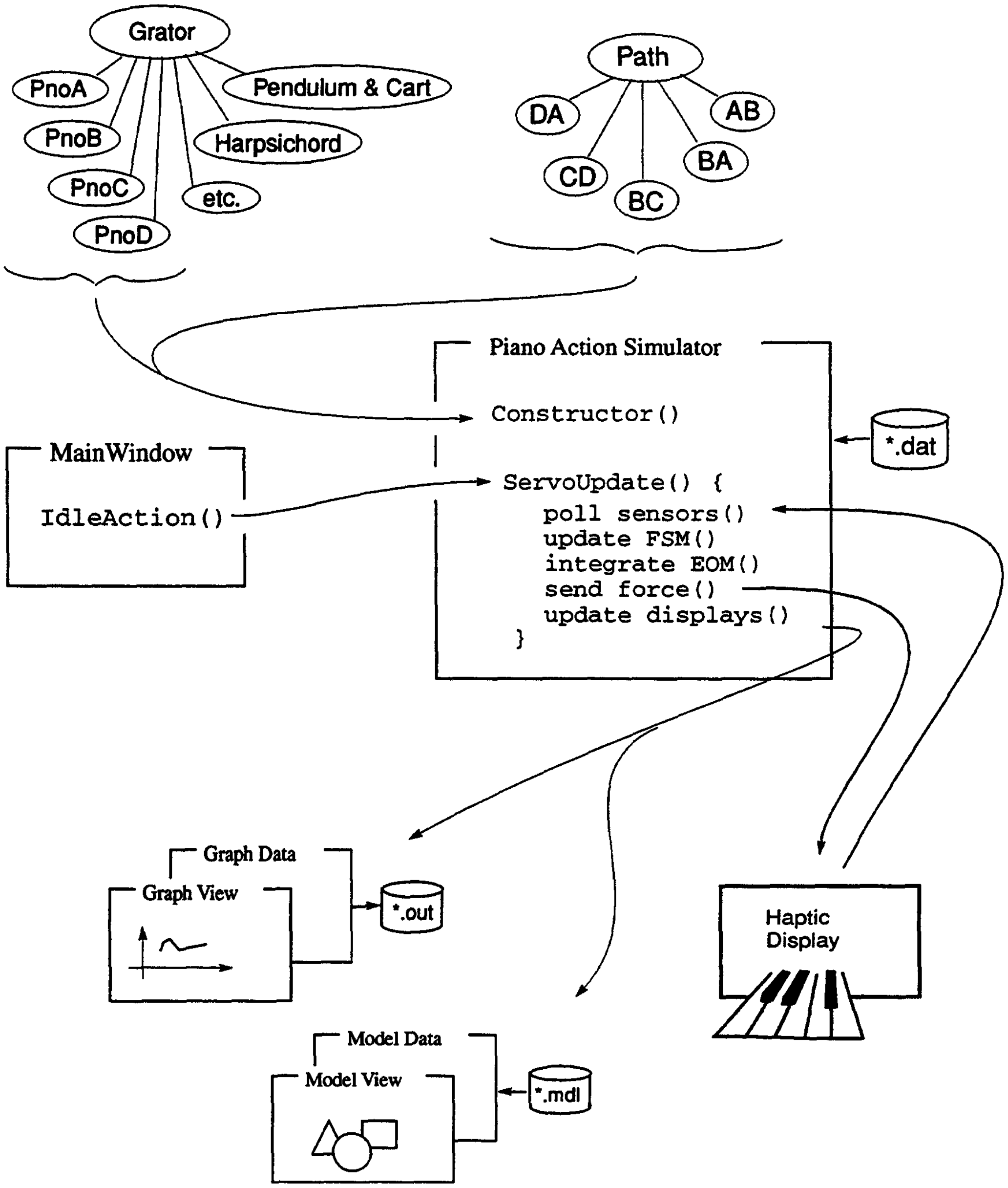
Readouts
Off

☒ Simulate ☒ Circulate
□ Animate □ Motomate
□ Audiate

Data
● Full

Scope
Ch A state2 Ch B state1

Figure 3.12: *Interface Devices for the Simulator*



```
Constructor()

ServoUpdate() {
   poll sensors()
   update FSM()
   integrate EOM()
   send force()
   update displays()
}
```

Figure 3.13: *Software Architecture for the Simulator*

3.6.3 Sample simulation output

In this section, the results of a simulation of the grand piano action model will be presented. A prespecified motion input of the manipulandum will be applied, and the response studied and compared to experimental results.

Values for the parameters L_1 through L_{27} which were used for the simulation are given in Table 3.3. Figure 3.14 shows the method which was used to determine values for each of the dimensions, and to determine the initial simulation values for the generalized coordinates. A plan-view video image of an isolated one-key piano action model (of the kind often displayed in piano showrooms) was digitized and imported into a *DRAW* program on the NeXT computer. Bold lines corresponding to the stick Figure were drawn over the image and a query tool available from the *DRAW* program was used to determine line dimensions in pixels and angles. These dimensions were then scaled to meters using certain known (directly measured) values.

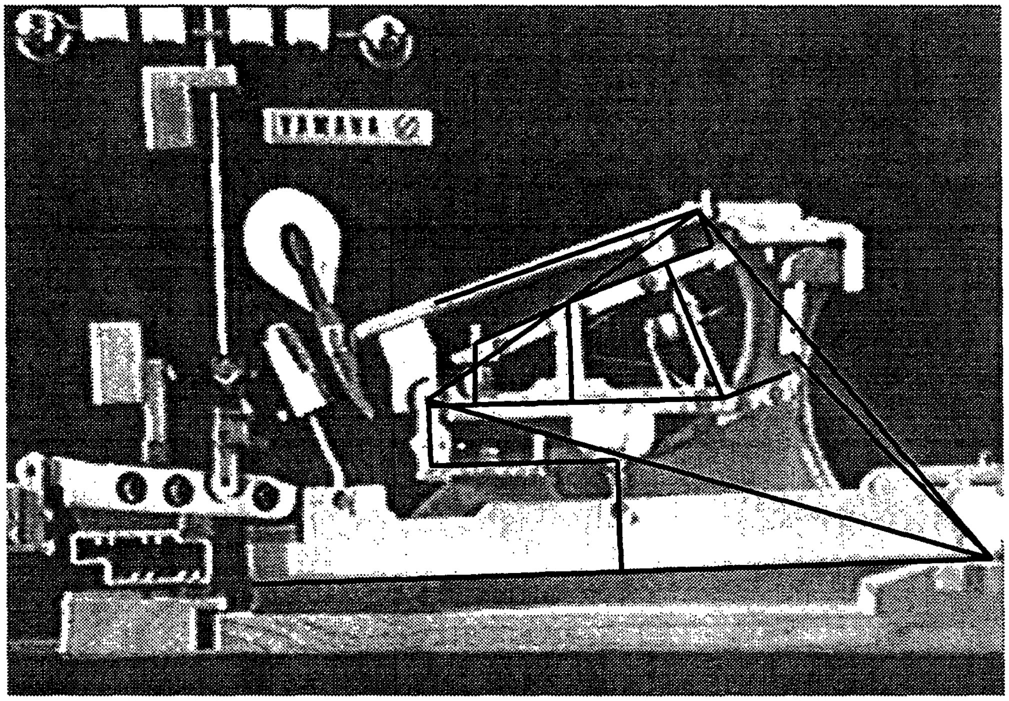


Figure 3.14: *Method for extracting certain dimensions and initial configuration angles from the actual piano action*

Table 3.4 gives the values used for the lumped parameters. These values were chosen for the

Table 3.3: Parameter Values

Dimension	Value [m]
L_1	0.0262
L_2	0.2286
L_3	0.1260
L_4	0.0386
L_5	0.0223
L_6	0.1001
L_7	0.0025
L_8	0.0250
L_9	0.0508
L_{10}	0.0129
L_{11}	0.0879
L_{12}	0.0170
L_{13}	0.0493
L_{14}	0.0366
L_{15}	0.0354
L_{16}	0.0351
L_{17}	0.0208
L_{18}	0.0556
L_{19}	0.0117
L_{20}	0.0988
L_{21}	0.0922
L_{22}	0.0544
L_{23}	0.0135
L_{24}	0.0521
L_{25}	0.0066
L_{26}	0.0064

most part by trial by error, although some guidelines were followed. Note that the stiffnesses used for spring-damper couplers (k_3 and k_6) are large. The 'force sensor' used for the J-H interaction force (k_4) was chosen quite high. The masses of the hammer and key are consistent with physically measured values. Damping coefficients were chosen generally by trial and error.

Table /refTab:InitialConditions shows the initial conditions used for simulation. These were derived as described in the text pertaining to Figure 3.14.

Figure 3.15 and 3.16 show the results of a simulation using the piano action model developed in the previous section. Figure 3.15 shows the trajectories of generalized coordinates q_1, q_2, q_3 and q_6 (the angles which the key, hammer, repetition lever, and jack make with the horizontal, respectively). Submodel simulation phases A, B, and C are also noted at the bottom of Figure 3.15. Also shown in Figure 3.15 is the driving input D (the angle which the manipulandum makes with the horizontal)

Table 3.4: Parameter Values

Stiffness Symbol	Value [N/m]	Damping Symbol	Value [N/m/s]	Mass Symbodl	Value [kg]
k_1	200.0	b_1	0.82	m_1	0.014
k_2	80.0			m_2	0.120
k_3	600.0			m_3	0.01
k_4	2000.0	b_4	0.2	m_4	0.05
k_5	0.35	b_5	0.06		
k_6	600.0				
		b_7	0.001		
		b_8	8.0		

Table 3.5: Initial Conditions

Generalized Coordinate	Value	Units
q_1	-0.03473	radians
q_2	-0.35622	radians
q_3	-0.38572	radians
q_4	0.0	meters
q_5	-0.03684	radians
q_6	-0.389557	radians
q_7	0.06223	meters
q_8	-0.0000254	meters
q_9 0.01649	meters	

used for this particular simulation run . Generalized coordinates D and q_1 are shown scaled by a factor of 4 to make their features apparent in Figure 3.15.

At time $t = 0.1$ seconds, the input D (manipulandum angle) begins to ramp up from -0.0335 radians. (During interactive simulation, D would be driven by the user, sensed by an encoder on the manipulandum). The ramp continues to $t = 0.18$ seconds and stops at -0.0055 radians.

Phase A acceleration

When the model simulation is initiated, a small amplitude transient is observed. This transient is due to the sudden application of gravity. As D ramps up, q_1 follows. Generalized coordinate q_2 (hammer) rises at a rate which about 5 times greater than q_1, consistent with the 5 times mechanical advantage which the key has over the hammer. Generalized coordinates q_3 and q_6 both decrease during phase A. Acceleration ends at $t = 0.1334$ seconds.

Phase B letoff

During letoff, q_1 continues to rise and q_2 rises at 5 times that rate. But now, q_3 and q_6 proceed in opposite directions. Generalized coordinate q_6 takes a sudden turn in direction at the transition from phase A to B. This point in time corresponds to the jack toe having met the escapement dolly. During phase B, by action of the constraint which prevents the jack toe from penetrating the dolly, the jack head begins to pivot out from under the hammer knuckle.

The end of phase B is signaled when the jack head has slid off the end of the hammer knuckle which occurs at $t = 0.181$ seconds. This event corresponds closely in time with the end of the ramp input on D, a chance occurance whcih was an effect of the particular choice of driving input for this model.

Phase C catch

At the beginning of phase C, when the hammer is first decoupled, the assembly of all elements but the hammer makes a somewhat abrupt move, quickly settling to its final configuration. This effect has to do with the sudden release of the action of damper b_8, which was impeding the sliding motion of Slider S_2 (between jack and hammer) during letoff.

At approximately $t = 0.2$ seconds, the hammer strikes the string (which is positioned at $q_2 = 0$ degrees) as seen by the reverse in direction of trace q_2. Thereafter, the hammer settles on the repetition lever. Low frequency oscillations are observed in q_2 and q_3. Very small amplitude oscillations are also evident in q_6 and q_1.

Indicator Functions

Figure 3.16 shows the traces of two indicator functions, g_{AB} and g_{BC}. Function g_{AB}, as the reader will recall from Table 3.5.3, is simply the vertical distance between the jack toe and the escapement dolly. When g_{AB} drops to zero, (which occurs at $t = 0.1445$ seconds) the transition from phase A to B is signalled. Function g_{AB} remains at zero (to satisfy the constraint associated with slider S2) during phases B and C.

Function g_{BC} is the displacement of slider S_3 (link between jack head and hammer knuckle) from the edge of the knuckle. When g_{BC} drops to zero, (at $t = 0.181$ seconds) the jack head has slipped out from under the knuckle.

Figure 3.17 shows the simulaed interaction force trajectories. The extension of the parallel spring-damper pair k_1-b_1 is simply $(q_7 - q_1) * k_1$. This is the force which would be displayed to the user through a hapter interface in an interactive simulation. The letoff phase corresponds to a period of increased force. This is the 'letoff resistance'. The trace of $q_4 * k_4$ is the interaction force

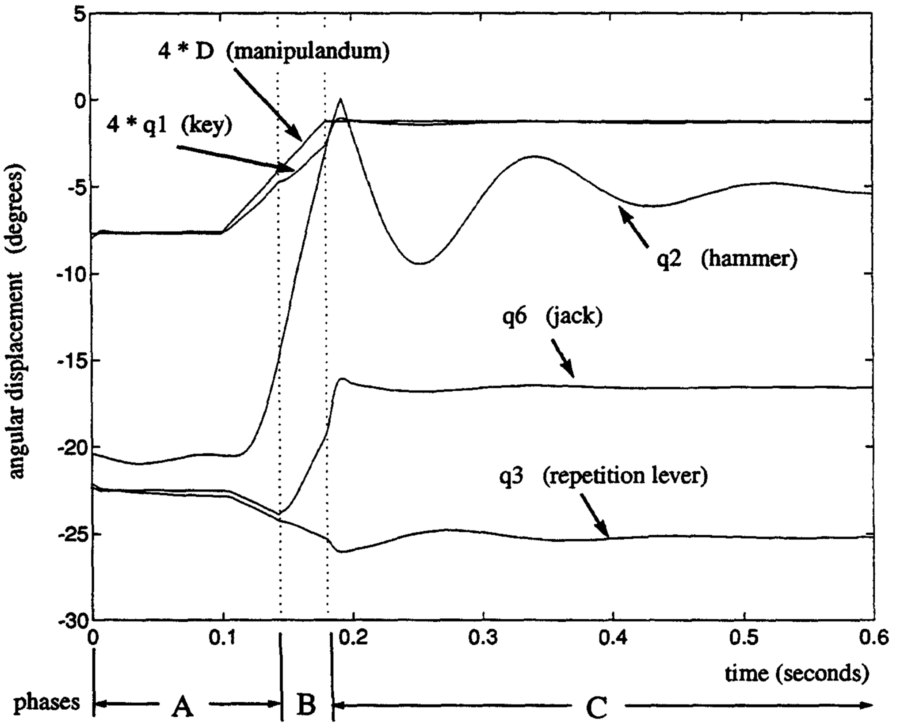


Figure 3.15: *Simulated Generalized Coordinate Trajectories*

between jack and hammer, along the axis of the jack. This force remained compressive throughout the entire simulation run, which is a function of the driving input. Had this trace wandered below zero, a release of the hammer would have occured by the 'other' $g_B C$ indicator function (see Table /refTab:IndicatorFunctions), $q_4 < 0$. Instead, release of the hammer occured by the jack slipping out from under the knuckle (q_9 threshold).

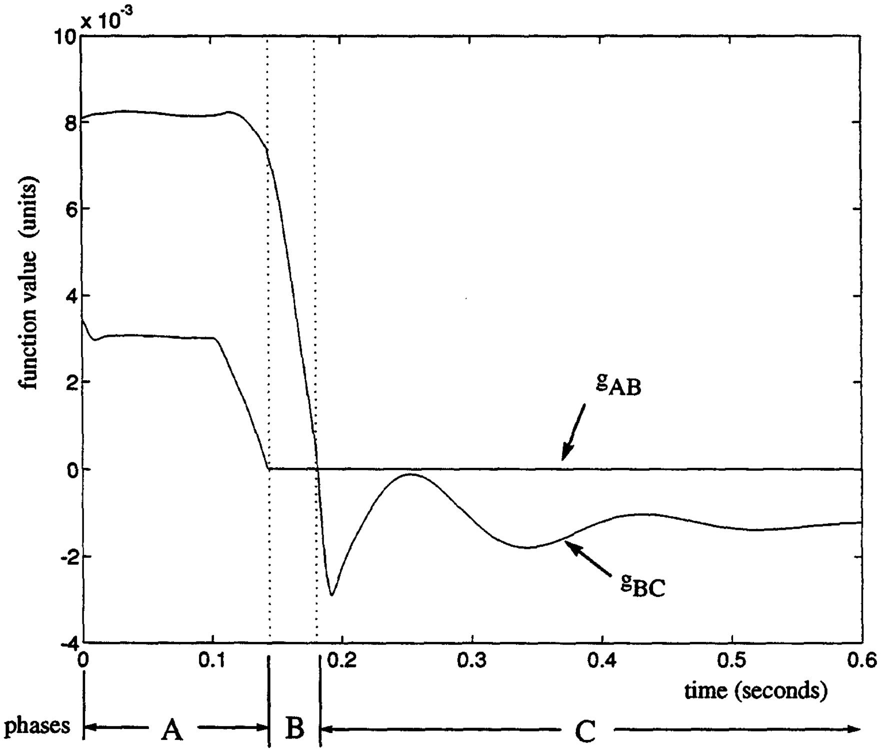


Figure 3.16: *Simulated Indicator Function Trajectories*

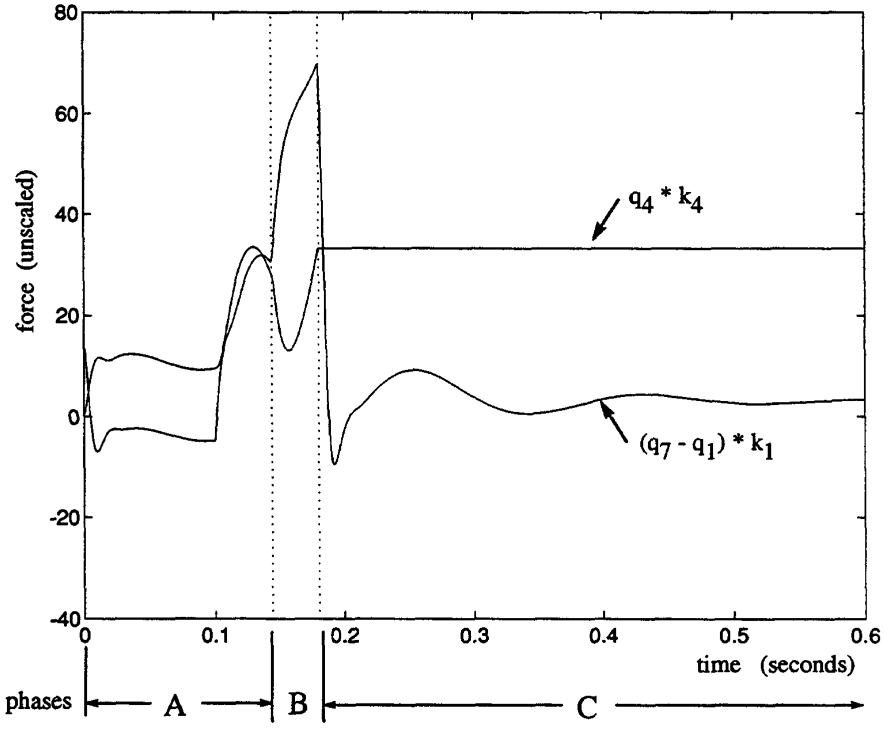


Figure 3.17: *Simulated Interaction Force Trajectories*

3.7 Experiment and Simulation Comparison

An isolated one-key grand piano action (see Figure 3.14) was used in an experiment to produce data against which the simulation output could be checked. The piano action was set into motion by releasing a weight from rest just above the key. The resulting motion of each action body was recorded using a high-speed video camera at 1000 frames per second. Retro-reflective patches were attached to ten locations on the piano action in order to facilitate vision recognition by computer. Illumination by bright lights and sensitivity adjustments on the camera produced an image for recording of 10 bright moving light patches on a dark background. Digitization and light patch centroid location determination from about 700 frames for each sequence was performed by Jim Walton of 4-D Video, Sebastapol, California. The digitized motions were used to deduce corresponding generalized coordinate trajectories with inverse trigonometric transformations. These experimentally determined generalized coordinate trajectories are shown in Figure 3.18.

As the key rises, the hammer rises at about 5 times that rate during the initial period of motion. Initially (presumably during acceleration) the jack and repetition lever (q_6 and q_3) both decrease. Then (presumably during letoff) q_6 and q_3 move in opposite directions. Finally, the hammer strikes the string (at $t = 0.2$ seconds into the recording) and then settles on the repetition lever, as seen by the oscillations in q_3 and q_2 after the string strike.

3.7.1 Discussion

Inspection of Figures 3.15 and 3.18 show strong similarity between simulated motion and experimental motion. But, as stated earlier, parameter values for simulation were chosen by trial and error to make the simulation results similar to the experimental data. It cannot, at this point, be claimed that the model will necessarily produce behavior indicative of its referent, when physically meaningful parameter values are chosen, only that similar motion *can* be produced by careful parameter selection.

Parameter selection can be an arduous process for a model with such complex behavior, but precisely because this model so closely follows the physical piano action in form, all adjustments may be expected to have intuitive (appropriate) effects.

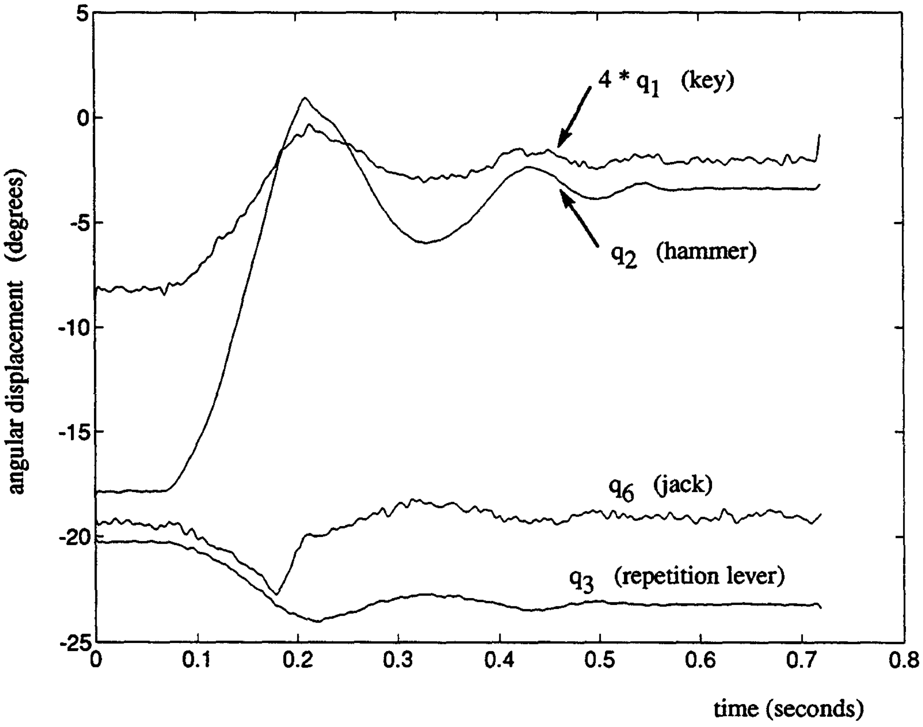


Figure 3.18: *Experimentally determined generalized coordinate trajectories*

3.8 Summary

An overview of the various model formulations for multibody dynamical systems revealed a tradeoff between ease of handling changing kinematic constraints and computational efficiency. Simulation through constraint changes of the most numerically efficient model form, the independent coordinates formulation, was identified as an area which has received little attention in the literature.

A modeling and simulation algorithm based on a model in independent coordinates has been presented which accommodates dynamical systems with changing kinematic constraints. Systems for which the various constraint conditions may be pre-determined but the ordering of constraints is left as a function of run-time conditions are handled. Submodels are constructed for the system in each of its constraint conditions in the independent coordinate formulation, providing for maximally efficient numerical simulation. An ODE solver and a set of submodel management routines in the form of a finite state machine are used for simulation, which may be run interactively, with multiple user input and output devices.

Various escapement mechanisms fit into this class of systems. A five-body model of the piano action was presented and used as an illustrative example of the simulator. This model is actually an hybrid of the coupled force balance and independent coordinate model formulations. The repetition lever uses coupling spring-damper pairs. The model does indeed exhibit behavior suggestive of its referent, the grand piano action, as shown through comparison of simulation and experimental data when parameters are chosen carefully.

Chapter 4

The Touchback Keyboard Design

The driving point mechanical impedance of a very large class of mechanical systems can be simulated using an ODE solver as the primary computational workhorse. In the following, I describe the construction of a haptic interface controller from an ODE solver. Other approaches will be reviewed to provide perspective.

4.1 Construction of a human-in-the-loop simulator from an ODE solver

The construction of an interactive simula*tor* from an off-line, non-real-time simula*tion* is relatively straight-forward. Those coordinates whose motion is *specified* in the dynamical model are sampled in real-time from the interface hardware rather than being read from an input file or calculated using a pre-defined function. Additionally, response forces from the model simulation which correspond to the same driving point as the specified coordinate are displayed in real-time through the haptic interface simply by commanding those forces to the actuator. This scheme is closely related to that used in flight simulators, which of course have been around for many years. Rather than motion or visual display, however, haptic display is concerned with making apparent the variable which is conjugate to the one sensed by the controller, and displaying that conjugate variable at the same driving point.

This construction can be implemented using an inverse dynamics simulation (forces computed in response to specified kinematic state). However, the forward dynamics simulation may be used if a degree of freedom is introduced in the model at the driving point through the addition of a coupling spring. Indeed, the coupling spring-damper pair which links the driven manipulandum to

Chapter 4

The Touchback Keyboard Design

The driving point mechanical impedance of a very large class of mechanical systems can be simulated using an ODE solver as the primary computational workhorse. In the following, I describe the construction of a haptic interface controller from an ODE solver. Other approaches will be reviewed to provide perspective.

4.1 Construction of a human-in-the-loop simulator from an ODE solver

The construction of an interactive simula*tor* from an off-line, non-real-time simula*tion* is relatively straight-forward. Those coordinates whose motion is *specified* in the dynamical model are sampled in real-time from the interface hardware rather than being read from an input file or calculated using a pre-defined function. Additionally, response forces from the model simulation which correspond to the same driving point as the specified coordinate are displayed in real-time through the haptic interface simply by commanding those forces to the actuator. This scheme is closely related to that used in flight simulators, which of course have been around for many years. Rather than motion or visual display, however, haptic display is concerned with making apparent the variable which is conjugate to the one sensed by the controller, and displaying that conjugate variable at the same driving point.

This construction can be implemented using an inverse dynamics simulation (forces computed in response to specified kinematic state). However, the forward dynamics simulation may be used if a degree of freedom is introduced in the model at the driving point through the addition of a coupling spring. Indeed, the coupling spring-damper pair which links the driven manipulandum to the virtual key introduced in the previous chapter serves this purpose. The interaction forces may be read as this spring's extension and the specified motion may be appropriately applied to the model by driving the coupled body.

Our haptic interface controller based,as described above on an ODE solver may be viewed as a kind of impedance controller, as depicted in Figure 4.1. To obey causality restrictions, the manipulandum must be viewed as an admittance and the human in turn as an impedance operator.

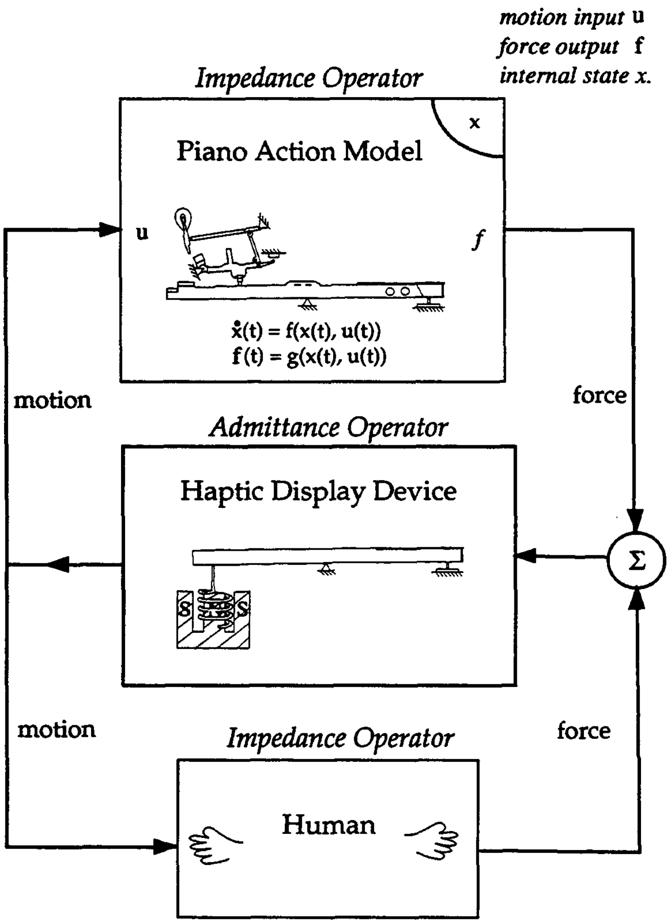


Figure 4.1: *Impedance Display through a Haptic Interface*

The impedance/admittance roles of controller and manipulandum may be reversed. In fact, the forward dynamics simulation may be used directly (without the coupling spring-damper pair) when forces are sensed from the manipulandum and kinematic state is imposed through the actuator. To impose kinematic state with an actuator, however, usually requires that an inner control loop be closed around the manipulandum. PID control can be used on the difference of desired and actual kinematic state, for example. See [64].

I prefer to call these implementations, in which an ODE-solver is involved, impedance *display* and admittance *display* rather than impedance and admittance control. But labels are somewhat hard to apply; it is usedful to relax definitions and explore overlap in display formulations. To further highlight the similarities and overlap, I will describe the haptic display of a simple sprung mass, as in Figure 4.2 using impedance control, impedance display, and admittance display.

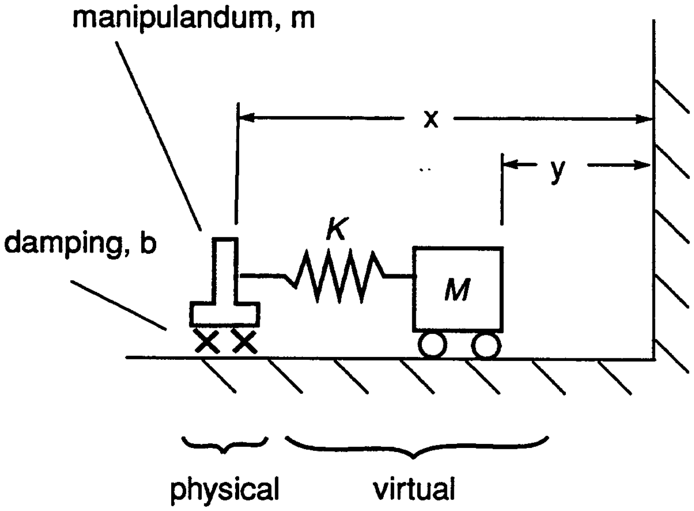


Figure 4.2: *Simple Model for Implementation in impedance and admittance display*

4.1.1 Impedance Control

Figure 4.3 shows the classic block diagram description of an impedance controller for haptic display. This diagram has been presented by Colgate in numerous papers [22], [25] [23]. $C(z)$ is simply a control law, usually $f = Kx_k + Bv_k$, which operates on sampled position x_k (and perhaps velocity v_k —not shown) to produce a force value for imposition on the manipulandum through a sample and hold and an amplifier. The manipulandum is modeled as a physical mass m coupled to ground through viscous damping of damping coefficient b.

It can be seen that impedance control is not capable of displaying the dynamics of a sprung mass. The operator $C(z)$ is memoryless, encoding no dynamics. It has no way to maintain the internal state y, the postion of the virtual mass.

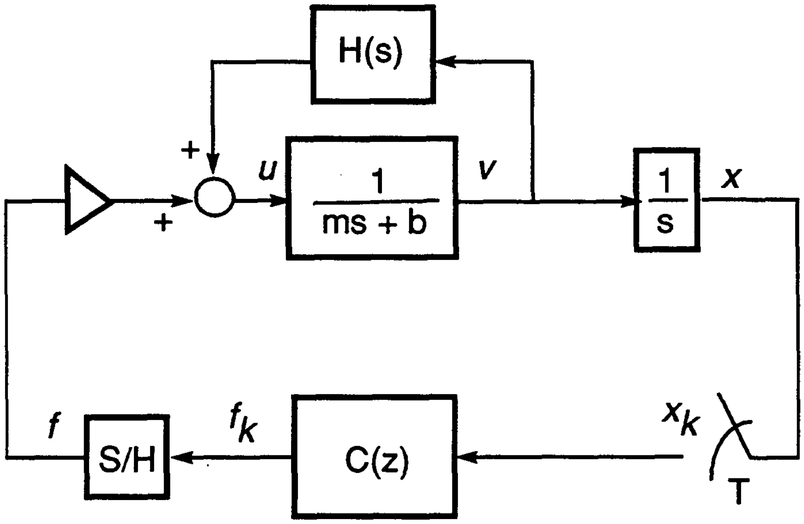


Figure 4.3: *The typical Z-Control Block Diagram*

4.1.2 Impedance Display

Figure 4.4 shows a detailed implementation of impedance display for a sprung mass. The sampled position x_k is differenced with the position y of a virtual mass, maintained by a numerical integration scheme. The basic formula for force display is: $f_k = K * (x_k - y)$. As this law is used to compute output force with each servo cycle, the solution to a second order model is computed with an ODE solver. In the simplest implementation, the state may be advanced through time as a function of input with the Euler method. The Euler method, applied to the equation of motion of a simple mass, $\ddot{y} = f/m$ reads:

$$\begin{aligned} v_{n+1} &= v_n - \tfrac{f_{n+1}}{m}\Delta t \\ y_{n+1} &= y_n + v_{n+1}\Delta t \end{aligned} \qquad (4.1)$$

where Δt is the step size or servo period. The spring K in this model corresponds to that degree of freedom which must be added to the model to allow a forward dynamics simulation. The depiction of numerical solution of differential equations with integration blocks in Figure 4.4 is somewhat awkward, but made to draw parallels to the next scheme, admittance display.

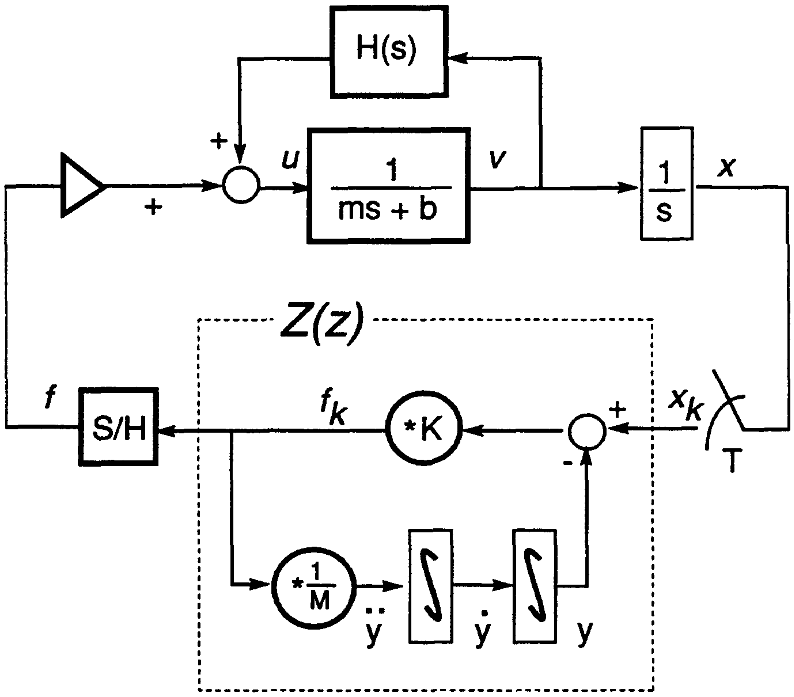


Figure 4.4: *The typical Z-Control (Impedance Display) Block Diagram*

4.1.3 Admittance Display

Figure 4.5 shows an implementation of a mass in admittance display. A force sensor is used to measure the interaction force f_h between manipulandum and human hand. This force is used in the simple constituent equation for a mass, $a = f/m$ and the resulting acceleration is integrated twice numerically to produce the position y. This position y is imposed on the manipulandum with proportional control (of gain K_p) in Figure 4.5. The gain K_p can be interpreted as a spring linking the simulated mass to the manipulandum. More generally, the position y is imposed on the manipulandum with a servo controller, such as a PID controller. See [64].

Note that in the above simple example of admittance control, the two integration blocks may be considered numerical integrators. In the impedance display implementation, the two integration blocks symbolize the numerical solution of a set of second order differential equations.

Boxes have been drawn and labeled Z(z) and Y(z) in Figures 4.4 and 4.5 respectively, to suggest why these schemes are sometimes called Z-control and Y-control. Rather loose interpretations of impedance and admittance are being applied in this case, since sampled position rather than velocity is involved.

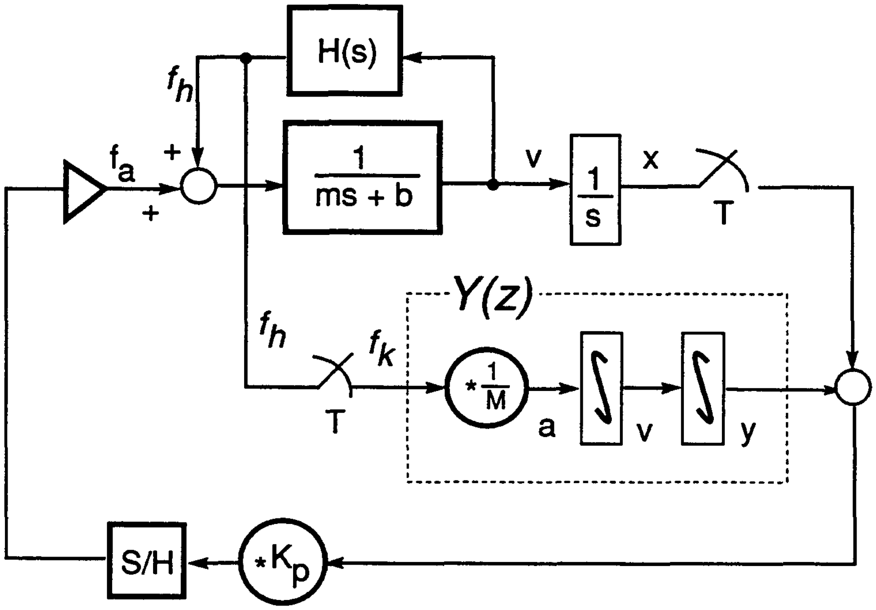


Figure 4.5: *The typical Y-Control (Admittance Display) Block Diagram*

4.1.4 Discussion

Issues to be considered in making a design choice between admittance display and impedance display include required sensors, noise sensitivity of those sensors, extensibility, maintenance, and so on. Supporting the choice between impedance display and admittance display (or impedance and admittance control) are surprisingly few guidelines and a rather small body of literature. The present-day lore is that the choice depends on whether the virtual environment impedance is dominated by inertia (in which case admittance control is recommended) or stiffness (in which case impedance control is recommended). Numerical properties, sensor resolution, and sensor noise must also be considered in these recommendations [64].

4.2 Design of the Touchback Keyboard

In this section, I will present the design of our Touchback Keyboard. The Touchback Keyboard is a haptic interface with a very particular application: re-creation of the mechanical impedance of the grand piano in a synthesizer keyboard. It thus takes on a particular form. The standard piano keyboard becomes is outer façade. It's inner design, however, is vastly different than that of the piano action. Each key is capstan (cable and pulley) driven by a small high quality ironless-core basket-wound motor and position sensored with an optical encoder.

Figure 4.6 shows an assembly drawing of the motor, pulley, motor mounting bracket, drum, and key mount. A highly flexible 0.012 in diameter steel cable with a 7x7x7 winding (not shown) couples the motor pulley to the drum. Note: hidden lines have not been removed in these drawings.

Perhaps the largest engineering challenge faced in this design was one of packing. Rather tight space restrictions inspired the eventual stacked and staggered design. Each key is mounted to a drum at one of seven angles. Figure 4.7 shows an assembly of four motorized keys. The drums are all bearing mounted to the same central shaft, but their radial placements around that shaft take on one of seven angles in a staggered fashion. Likewise, the drive components associated with each drum take on those same seven angles. In the end, all keys point forward and all motors, mounting plates, cables and other components fit neatly without interference. The keymounts rotate through small angles on the central shaft while the motor mounts are held securely in place by a box housing (not shown in the drawings). After seven keys, the arrangement may be repeated, for the width of seven keys is just over the length of one motor.

Our final selection for the mechanical advantage from motor to key was 24:1, which was based on a tradeoff of torque capacity against intertia as discussion below. 512-count per revolution encoders were incorporated with an additional advantage over the motor of 8:1, thus providing over 5,000

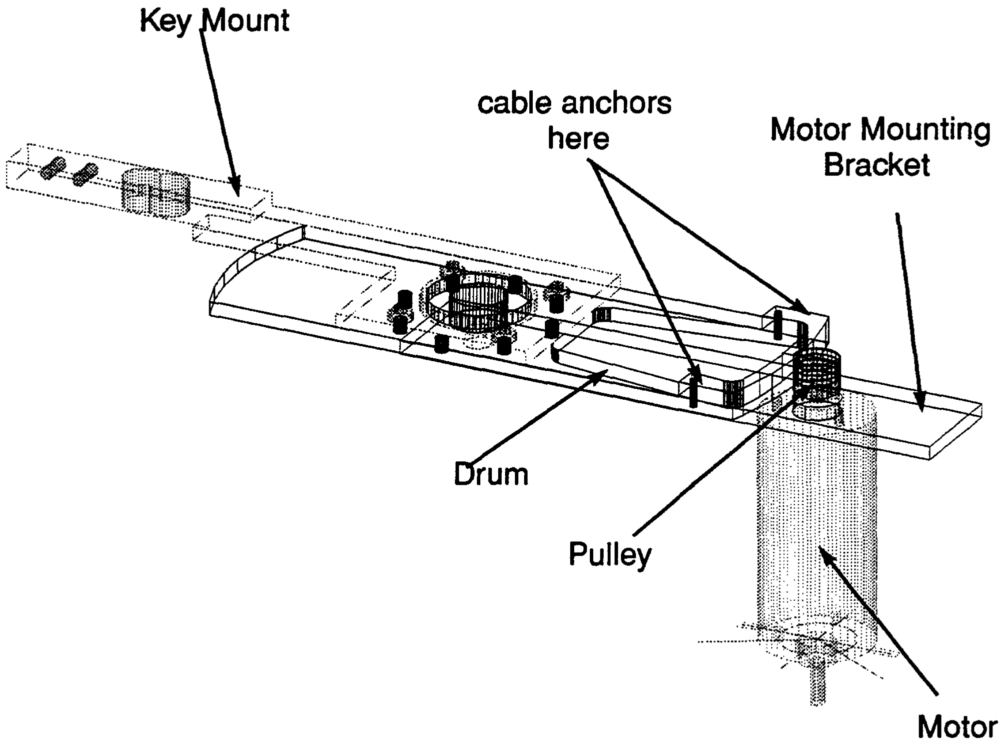


Figure 4.6: *One Key Assembly Drawing*

counts of encoder resolution (with quadrature counting) for the five degrees of key motion. This fine resolution was used to allow numerical differentiation of the position signal in lieu of a tachometer.

We have aimed to create a device which in its unpowered state has the mechanical properties of the key alone but when powered may be made to take on the impedance properties of the full piano action. All elements of the piano action apart from the key are to be rendered through the workings of the motor.

The determination of the target unpowered inertia was made with simple empirical studies on the inertia of the key (a bifilar pendulum experiment). The target force output capabilities were determined using simple models of the piano action (like those presented and reviewed in the previous chapter) and experimental data on piano playing forces available from the literature. The mechanical advantage of the motor was carefully sized to trade off reflected unpowered motor inertia to maximum force output.

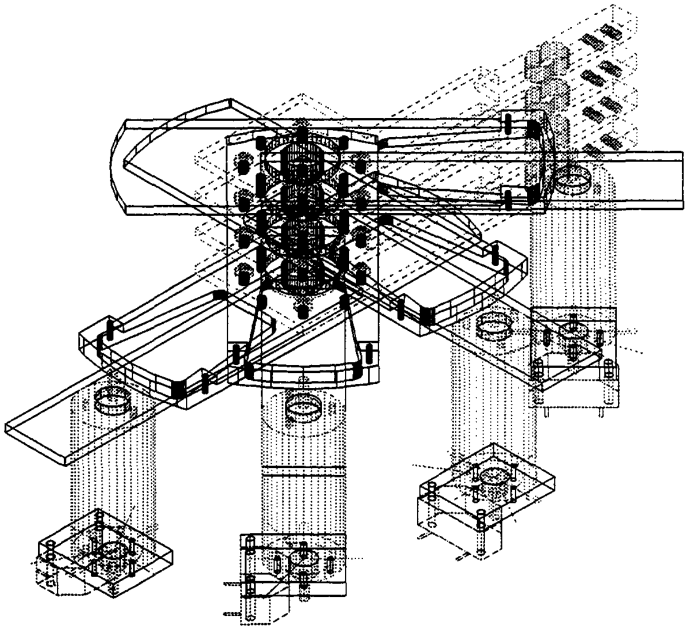


Figure 4.7: *Four-Key Assembly Drawing*

Figures 4.8 through 4.11 highlight the design in photographs. The electrical cables connecting each of the encoders and the motors to the computer are apparent in Figure 4.8. A cubical box with two open sides houses the assembly and secures each motor mount to its appropriate angular position. Figures 4.9 and 4.10 further document the mounting of each motor assembly to the box. Especially in Figure 4.10, looking down the central shaft, one sees that all but one of the eight possible 45-degree staggered positions is occupied by a motor assembly. The one angular position left out is occupied by the plane of keys itself. Figure 4.11 shows a view from above with the top plate of the box removed. The tight packing of all plates and motors can be appreciated in this view. The leftmost key features a mounted straingage and amplifier circuit for force sensing. Each of the key mounts are outfitted with a binocular-type strain concentrator, located at the area of mounting of the plastic key. So far, only the lowest key has been fully instrumented.

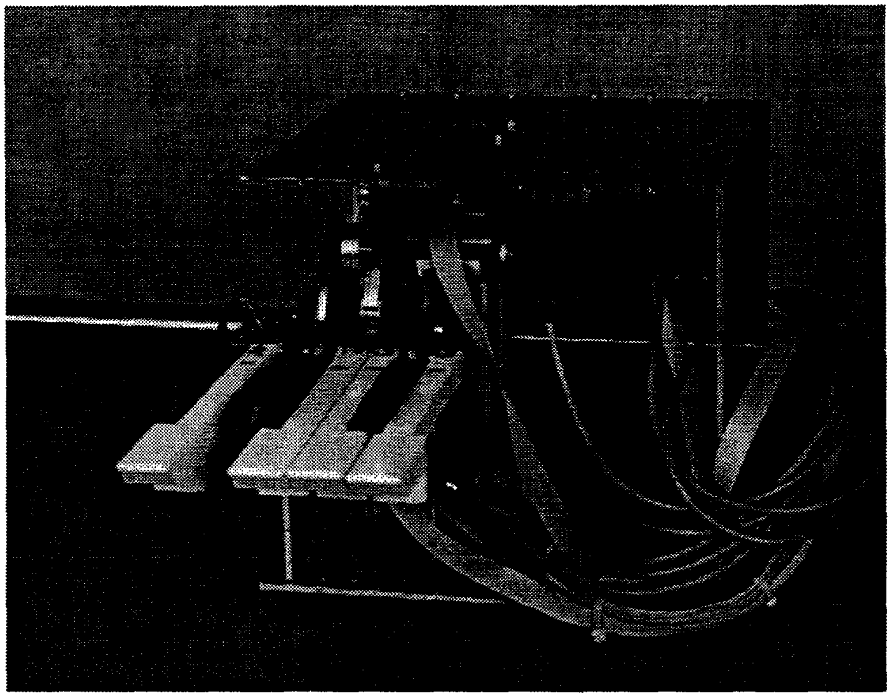


Figure 4.8: *Touchback Keyboard: Front View Showing Cables*

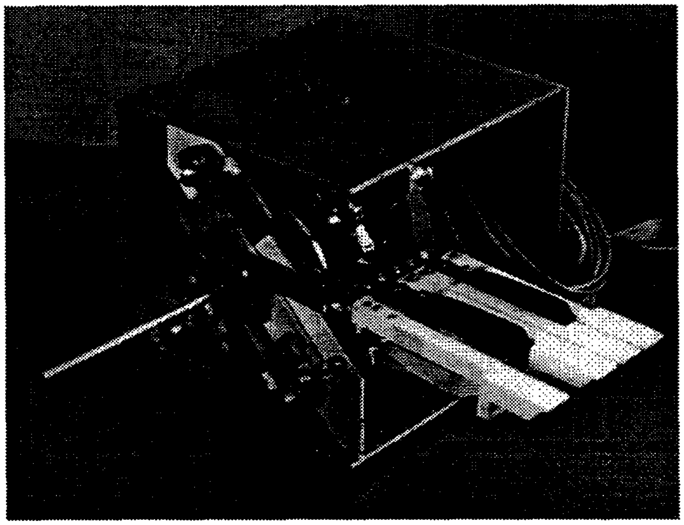


Figure 4.9: *Touchback Keyboard: Isometric View*

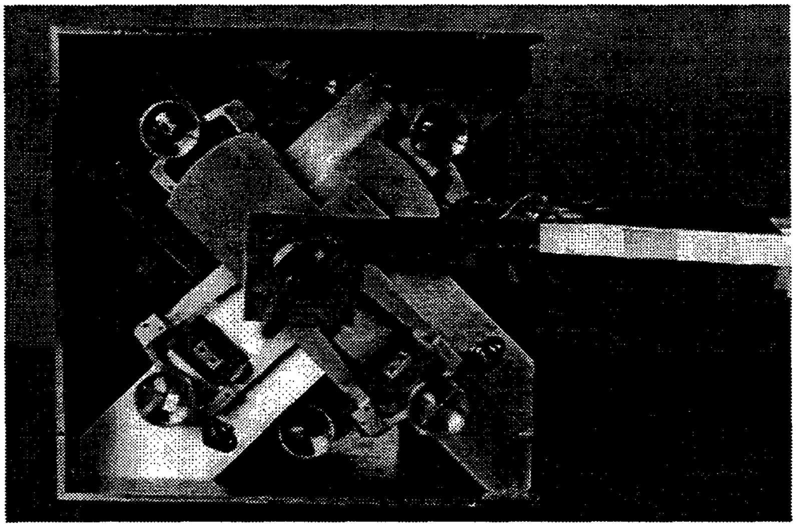


Figure 4.10: *Touchback Keyboard: View Looking Down Skewer*

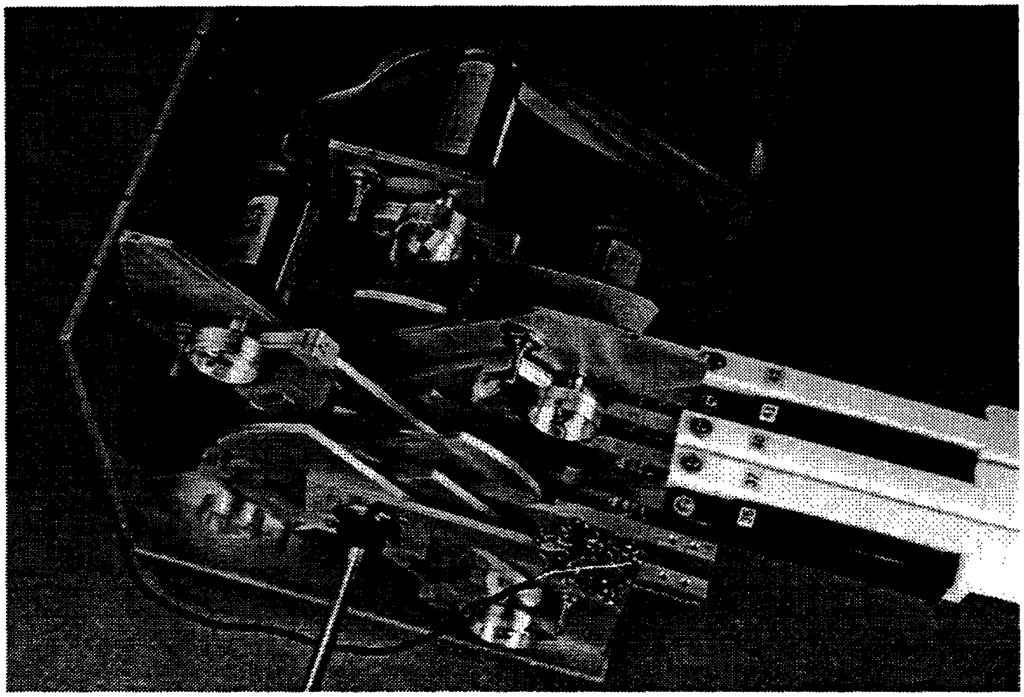


Figure 4.11: *Touchback Keyboard: View with Top Open*

Chapter 5

Passive Rendering of the Virtual Wall

5.1 Introduction

5.1.1 Motivation

The virtual wall is the simplest example of a programmable boundary within the workspace of a manipulandum. As such, the virtual wall is a fundamental component of almost all virtual objects. It is set up with an *if* statement:

> *if beyond a certain position, react; else do nothing.*

The wall is the incarnation in ***virtual*** reality of a unilateral constraint. The wall comprises a simple configuration-dependent changing kinematic constraint. As discussed in previous chapters, such changing constraints are an important consideration in rendering the piano action for haptic display. The fact that the wall fully yet simply encompasses the notion of changing constraint conditions or changing sub-models makes it a natural and worthy topic of study in this thesis. In the present chapter, the virtual wall will be studied systematically with the aim of developing robust algorithms for simulation across constraint discontinuities.

Beyond its position as a fundamental building block of virtual objects, the virtual wall rouses research interest because of the difficulties which its realization presents in practice. Despite its apparent simplicity, the virtual wall usually evades perceptually convincing renderings. When touching a wall, especially one which is meant to be stiff, undesirable vibratory motion of the manipulandum (often called contact instability or chatter) tends to arise. The manipulandum continually makes and breakes contact with the virtual wall during this behavior. Such non-passive behavior immediately expunges any sense of immersion which the human operator may have been enjoying prior to encountering that chattery wall.

Because it is such a challenge to implement chatter-free, the virtual wall is already finding use as a benchmark for performance comparisons between haptic interfaces. Thomas Massie quotes a maximum wall stiffness attainable using his PHANToM haptic interface of XX N/m, where the criterion is presumably the non-existence of chatter [65]. Colgate has proposed the range of passively displayable impedances as a useful measure of attainable performance by a haptic interface, which he calls Z-range [22]. However, the wall, because of its discontinuous nature which makes it a bigger challenge, should be considered in the suite of objects which can be rendered passively by a haptic interface.

Outline

This chapter will address the problem of chatter associated with stiff virtual walls by developing improved controller designs. These controllers (or virtual wall algorithms), when used in the standard digital implementation for haptic display, will render walls which do not suffer chatter even when the wall stiffness is high and the sampling period long.

In the remainder of this introduction, I will discuss the origins of chatter in the virtual wall. The roles of the human, the haptic interface device, and the controller in the mechanisms whereby mechanical energy is introduced (which exhibits itself as chatter) will each be detailed. Various factors may underlie a tendency toward unstable behavior observed in a controlled, coupled system such as the virtual wall. These include non-colocated sensor and actuator, system dynamics which are unmodeled or otherwise omitted from the controller design, and sensor signal quantization. Some of these mechanisms can be avoided by informed mechanical design, others are more difficult to avoid. Two culprits which are not easily quelled by good design will be identified and singled out for analysis in this chapter. First, the zero-order-hold operator and second, the possible asynchrony of the wall threshold crossings with the sampling times. Both are inevitable consequences of the sampled data implementation of the virtual wall. This introduction will wrap up with an enumeration of claims about two new controllers to be presented which compensate for the ill-effects of the zero order hold and intersample threshold crossing. Section 2 will review the literature pertaining to controller design for haptic display, especially with regard to virtual walls. In section 3, the model of a bouncing ball which will serve as a useful allegory for the development of the controller designs is presented. In section 4, the design of the two improved virtual wall algorithms will be carefully developed. The first design uses model-based prediction, the second design makes use of standard state-space digital control design techniques. Section 5 will present results from an experimental implementation of both of these candidate virtual walls. Section 6 will discuss and summarize and Section 7 will present extensions.

The next chapter will present a thorough analysis of virtual walls with and without the improvements introduced in the present chapter. The goal in the next chapter will be to produce measures useful for gauging and predicting the performance improvements which result when these new controllers are implemented. Treatments in the next chapter will be of a more theoretical nature and another literature review section will be included.

5.1.2 Origins of chatter in the virtual wall

Assumptions regarding the role of the human

The tendency of chatter to arise is naturally a function of the wall algorithm with its parameters and the physical properties of the haptic interface, but this tendency toward chatter also depends to a large extent on the physical properties of the human user—specifically, the human's driving point mechanical impedance at the interface. (Throughout our treatment, we assume that the human's finger maintains contact with the manipulandum.) The dependence on the human's impedance is not so surprising when we realize that under consideration is the *interaction* behavior of two dynamical systems, the manipulandum and the human limb. But even further, under consideration is the interaction behavior of two *controlled* dynamical systems. Behavioral predictions cannot be made until both systems (manipulandum and human), each with their controller (computer and brain) are brought into the analysis. For example, note that the driving point impedance of the human hand or finger can be modulated (within certain bounds) by the human operator by changing muscle activation levels or by changing hand/finger postures. Thus, by pressing in certain ways, chatter against a virtual wall can be selectively induced and sustained, and sometimes even amplitude-modulated. Another interesting empirical observation to be made regarding walls and the human exploring them is that the same wall may be destabilizable (prone to chatter) under the fingers of one person while always remaining stable under the fingers of another. Presumably this effect is due to the differing impedance properties of the fingers of the two explorers.

The foregoing examples highlight the way in which chatter is usually encountered and points to an important modeling assumption which can be used to greatly simplify the analysis (and design) of the virtual wall—that is to assume constant command on the part of the human. Chatter frequencies are typically 10-30 Hz. An effective strategy for the human to induce chatter is usually not to move back and forth at high frequency but rather to adopt and maintain a certain impedance while simply hitting the wall a single time (or even gently coming up against the wall). Although not always beyond the command capabilities of the human, typical chatter frequencies are certainly high compared to the frequencies which characterize the wall-strike *intentions* of the human. Therefore, assumptions of constant control output by the human will be made in the following analyses and control designs. Assumptions about the particular human impedance and bias-force command level itself will be left open for as long as possible. The primary culprit in the chattery interactive behavior is assumed to be the discontinuous and digital nature of the manipulandum controller (the digitally implemented virtual wall algorithm) rather than the effects of any varying command from the human. Armed primarily with the observation that chatter remains, both empirically and in simulated settings, when a constant impedance is adopted by the human, we assert that the human is not responsible for introducing energy into the system. We will assume that the human can be modeled with a constant, passive impedance. [1]

In fact, we will assume that the wall-exploring human may be fit with a second order linear time-invariant model and draw justification for this assumption from two items. First we note that contact instability in a virtual wall does not depend on time variance of the human as mentioned above–the problem remains when the human wall explorer refrains from making volitional movements. Second, the literature indicates that second order linear models may be used to model a human finger to a very good degree of fit [39] (and references contained therein), so long as the time durations are short.

The sampled data system

The virtual wall is commonly rendered for haptic display using the very simple algorithm given in Table 5.1. [2]

In natural language, if the sampled position of the manipulandum y_k is beyond a setpoint y_{wall}, exert a restoring force f_k proportional to its distance beyond that setpoint, else do nothing.

Occasionally damping is also used as in the following control law:

$$f_k = K(y_k - y_{wall}) + Bv_k \tag{5.1}$$

[1] We adopt the standard definition of passivity, that energy cannot be extracted from a passive system. Technically: for all possible force/motion trajectories, the running integral of force time velocity is always less than the initial stored energy.

[2] Note that the pair of *if/else* logical statements in the Pseudocode of Table 5.1 may be replaced by the single statement $if(y > y_{wall})\ y = y_{wall}$, placed before the evaluation of the control law. Such a code structure would be more suggestive of the unilateral operator used in the block diagram figures which follow shortly within this discussion. The *if/else* structure, however, is more suggestive of a switching control law, conditioned on the sampled position. Subtle distinctions *do* arise for damped virtual walls when a first-difference approximation is made for velocity from the sampled position, and that approximation is considered to be part of the control law. For our purposes there is no difference between hese two code structures.

Table 5.1: **Pseudocode for the Virtual Wall**

```
loop at sample rate {
    sample manipulandum position $y$.
    if $y_k < y_{wall}$
        $f_k = K(y_k - y_{wall})$;
    else
        $f_k = 0.0$;
    display force $f_k$
}
```

where v_k is the sampled manipulandum velocity. If the manipulandum velocity is not available from a sensor, the position is numerically differentiated to produce v_k. The loop is typically run at servo-rates of 300 to 1000 Hz. Hysteresis is sometimes added in the hopes of dispelling chatter and further improving perceived hardness. These and other approaches, successful to various degrees, will be discussed in the Literature Review section to follow.

The above algorithm is implemented as the digital controller $C(z)$ in a sampled data system as shown in Figure 5.1. The haptic interface device is shown as the continuous plant $P(s)$ and the human as the continuous linear impedance operator $H(s)$. This sampled data system naturally comprises both continuous and discrete elements, linked through the Sample and Hold operator S/H and the sampler of period T, as shown in Figure 5.1. The Sample and Hold operator is responsible for sampling the controller output (at sampling times which are assumed to be synchronous with the sampler T) and holding these constant until the next sampling time (zero order hold). A unilateral nonlinearity, symbolized with an icon of its graph, is used to encapsulate the action of the $if/else$ switching statements in Figure 5.1.

Now that the elements of the virtual wall have been laid out and the human has been acquitted of the energy-introduction crime, it is time to implicate the real offenders. The offenders are the zero-order hold and intersample-threshold crossing. First we discuss the role of the zero-order hold.

Origins of non-passive behavior: the zero order hold

Though the virtual wall implementation does have elements capable of giving rise to active behavior (the motor and motor amplifier), these are not directly responsible for chatter in the virtual wall. If the motor produces only those reaction forces which mimic the reaction forces of a physical unilateral spring, the controlled motor would appear passive, despite the fact that its amplifier is plugged into

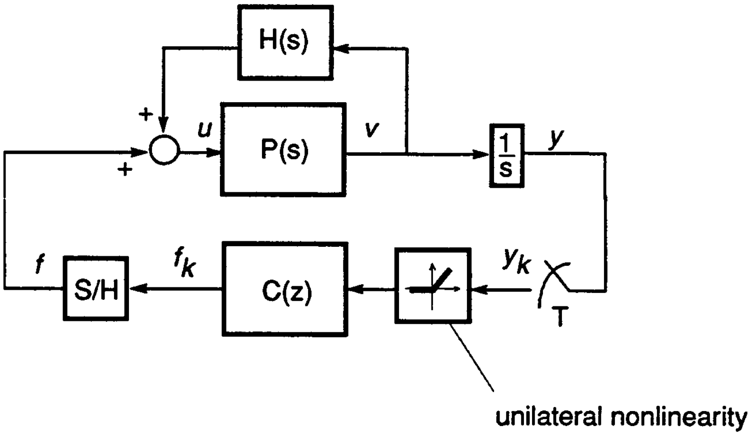


Figure 5.1: *Implementation of a Virtual Wall*

the wall. We cannot assume, however, that a discretely implemented but continuous-time inspired spring-damper control law will cause the motor to behave passively. The offense has been committed while implementing in the discrete domain a wall designed in the continuous domain. Specifically, the sample and hold (zero order hold) operator and the possibility of the crossing of the wall threshold being asynchronous with the sampling times can be identified as the means of introduction of energy.

It is well known in digital control theory that a controller designed in the continuous domain will yield poor results if implemented in the digital domain as a sampled-data controller when the sampling period is long. The standard rule of thumb used in digital control design requires that the sampling rate be 20 times the highest expected system frequency. If this guideline is not met, the closed loop system will likely be effectively destabilized by those designs which are produced with continuous system methods. Now, given sufficient physical damping or (to a possibly different degree) positive virtual damping, the closed loop system may nonetheless exhibit passive behavior. However, such requirements on the design can be considered sub-optimal since they require increased actuator authority. More discussion on this topic will take place in the Literature Review section to follow.

Intuitive explanation

An intuitive explanation of the energy-producing effects of the discrete (sampled-data) implementation of a wall created with the control law for the undamped wall,

$$f_k = K(y_k - y_{wall}) \tag{5.2}$$

is now offered. This description roughly follows that of Colgate in [23].

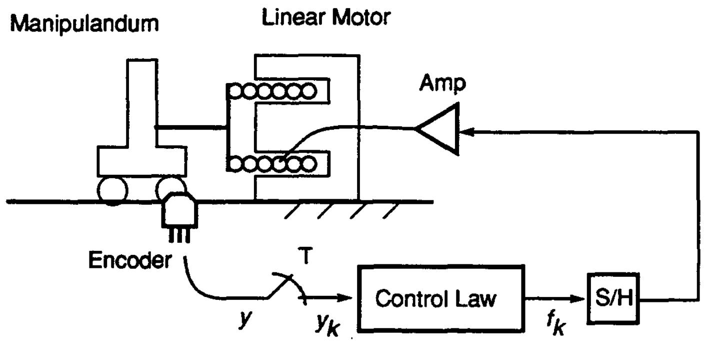


Figure 5.2: *Implementation of a Virtual Wall*

Figure 5.2 shows a very typical virtual wall implementation, into which the control algorithm of Eq. 5.2 would be inserted in the control block. The position of a linear single-axis backdrivable manipulandum is sampled by an encoder (with assumed infinite spatial resolution) and used as y_k in the algorithm. The algorithm output f_k is zero-order held before being amplified and used in the form of current to drive the linear motor attached to the manipulandum.

While moving into the wall, the sampled manipulandum position will necessarily (except at the sampling times themselves) lie closer to the virtual wall surface than the actual position of the manipulandum. Consequently, the force output, while moving into the wall, will (except at the sample times) be lower than it would have been for a continuous wall. By contrast, while moving out of the wall, the sampled position will lie deeper inside the wall (except at the exact sample times, where it is correct) than the actual manipulandum position and consequently the force will be (by comparison to the real wall), too high. Thus as one presses on the virtual wall, one needs to perform less work than one would on the real wall (its referent) to produce the same deformation. As one lets go, one has more work returned by the virtual wall than would have been returned by its real-world counterpart. Thus to simply push on the wall and let go (a common exploratory procedure) is an effective method for extracting energy from the wall. The virtual wall is, obviously, quite non-passive.

Figure 5.3 shows a trace of the zero-order-held force output history versus the position history which produced it overlaid on a graph of the constituent equation $f = Kx$ of the referent wall of

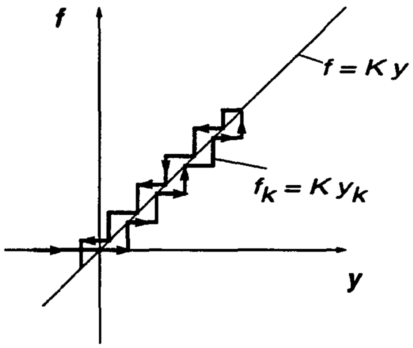


Figure 5.3: *Tracing out the Virtual Wall*

stiffness K. From such a plot, it can easily be seen that the time-average force while moving into the wall is by comparison to the constituent equation too low and on the way out too high. A *negative* hysteresis curve is thus introduced which produces energy when traversed.

Origins of non-passive behavior: asynchronous switching times

From the intuitive discussion above, it is apparent that the zero-order hold is responsible for non-passive behavior of virtual walls, but I will point out a second slightly more subtle energy-instilling aspect of the digital implementation of the virtual wall. This factor only plays a role in unilaterally acting virtual objects such as the wall, it is not present in objects which lack changing constraint conditions.

Briefly, this effect is due to asynchrony of would-be constraint changes in the virtual wall with the sampling times of the controller. Constraint changes should occur when the indicator function (as defined in previous chapters) evaluates to zero. However, because of discrete sampling and the ZOH, changes are not enacted until the next sampling time after the trip of an indicator function. Crossings of the threshold (trips of the indicator) which occur between sample times can effectively create energy.

Considering a wall of stiffness $K = 1$ allows us to view the control signal in an informative time-chart. Figure 5.4 shows the trace of the zero-order held output force resulting from a single strike of the unit-stiffness wall overlaid on top of a time trace of the manipulandum position y. This plot is somewhat idealized, but experimental plots are very similar. See Figure 3.18 for a plot of the virtual wall reaction force overlayed on the manipulandum position derived using a virtual wall implementation.

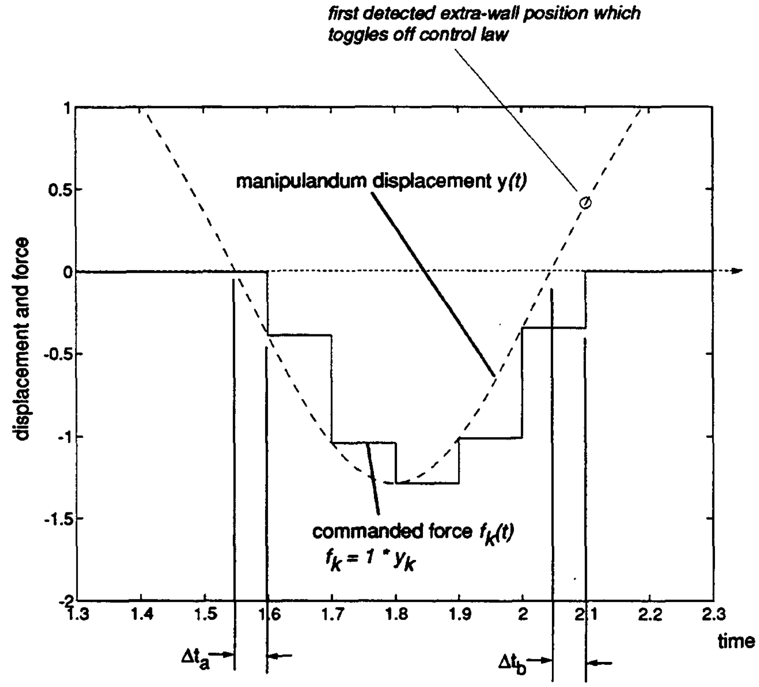


Figure 5.4: *Time-chart of modeled manipulandum position and control signal*

As is depicted in the force-displacement plot of Figure 5.3, the on-average small valued force on the way into the wall and large valued force on the way out of the wall can be seen. But two time periods in particular are highlighted. The first, labeled Δt_a, is the delay in turning on the wall controller and Δt_b is the delay in turning off the wall. If it were not for the discrete implementation of the algorithm, whether the wall was on or off would be strictly a function of configuration. However, because of discrete, constant step size sampling, the switching times become a function of both configuration and time. Because the crossing of the configuration threshold will not in general occur on the sample times, yet model switching is only admitted on the sample times, errors will be committed. For example, the wall will likely first be turned on with the wall's spring in

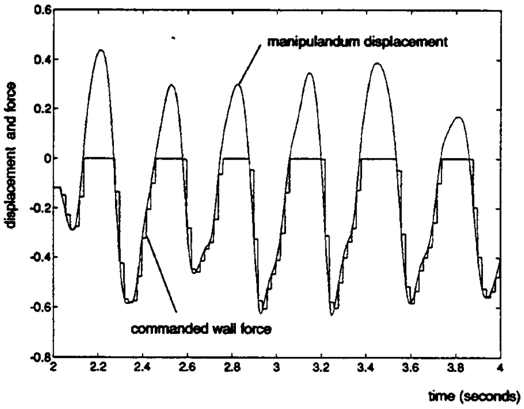


Figure 5.5: *Time-chart of experimental manipulandum position and control signal*

a slightly compressed state because the first sampled position to trip the conditional will not be located precisely on the boundary. Such an error can produce energy since the spring now stores energy without the requisite work having been done on the spring.

Upon leaving the wall, the first sampled manipulandum position will, in all likelihood, lie outside of the wall rather than just on the threshold. In this case, since the wall will immediately be turned off, (assuming no computational delays) by the algorithm of Table 5.1, the wall will no do extra work on the human (the force for the last wall-on sample period will still push away from the wall). Thus we see that the asynchrony of the wall on/off switching times with the sampling times are energy-producing. [3]

5.1.3 Claims

Interaction with virtual objects through a haptic interface imposes two conditions on the simulation algorithm, namely that the simulation be run with a constant step size and that the discrete simulation output be zero-order held before acting through the motor on the interface. These two requirements are simple consequences of the fact that the haptic interface and the human are continuous systems and we would like to implement the virtual object dynamics with a digital computer,

[3] Note that this asynchrony can alternatively be interpreted as misalignment in space of the on/off switching points with the wall location (threshold) itself. These two interpretations are both valid, but neither provides more affordances to the solution since in the final analysis, the switching effects are a function of both position and (sampled) time. A worst-case or averaging assumption must be made before one effect can be treated independently of the other, as will become apparent in the next chapter.

making the whole a sampled data system. The goal is to have the apparent dynamics of the haptic interface change in response to varying configuration of the virtual environment just as the dynamics of a real wall change in response to varying configurations. That is, the wall algorithm should execute without any dependencies on either the step-size or the relative placement of discontinuities and sampling times, despite the sampled-data nature of the linking of the discrete algorithm (simulation) to the continuous interface and human.

We have developed two controllers which render, in sampled data settings, passive virtual walls. The first of these is more easily generalized to other virtual objects, for it approaches the problem from the simulation perspective (to be defined shortly). The second controller is the simpler of the two, but it is not quite as extensible. Its design makes use of tools from digital control.

Both designs begin with modeling assumptions about the human operator and the haptic interface device, as motivated by the discussions above. In the case of the first design, these models are used to make predictions of behavior one half sample ahead so that the approximate half-sample delay of the zero-order hold can be effectively cancelled. Such strategies have been used in flight simulation with motion display [48]. Additionally, these models are used to derive the state at the inter-sample threshold crossing times from state information at the sampling times. These inter-sample states are used to synthesize special control signals which stay within the constraints of their sampled-data implementation yet drive the system to the state appropriate in the corresponding continuous-time system. This is an application of 'deadbeat control' techniques which are available in digital control design but not in continuous control design. Thus the errors of inter-sample switching are fully accounted for.

In the case of the second design, models of the human and the haptic interface device are directly incorporated into a controller design process which takes place in the digital domain. A zero-order hold equivalent model of the human and device is used during the design of a digital wall algorithm (controller) which, both when simulated in the digital domain and implemented in the sampled data system, yields the desired results. Finally, to handle behavior irregularities due to possible intersample switching times, a method identical to that used in the first design (deadbeat control) is used to correct for errors commited becuase of intersample threshold crossing.

Simulator versus Controller

We use the term simulator somewhat interchangeably with the term controller in the present context because both words apply. The digital computer is on the one hand a controller, making the haptic interface exhibit dynamics (in response to user input) which are not its own, and on the other hand a simulator, responsible for maintaining and advancing in time (in response to user input) the state of a virtual object. For the simulation of a static (memoryless) object such as the virtual wall, use of a state is not necessary, the computer need not encode any dynamics, and the word controller is more appropriate. For a dynamical object, however, a numerical simulation scheme must be used, unless a solution of its equation of motion is available. Certain simulated variables are interpreted as control output and certain others are fed in real-time into the simulator, making its interpretation as a controller complete. The methods developed in this chapter for the virtual wall will be extended to more complex dynamical systems using the simulator viewpoint.

5.2 Literature Review

5.2.1 Robotics Literature

Contact instability observed in the virtual wall is closely related to contact instability observed between a robot, especially a force-controlled robot, and its environment. The destabilizing factors at play in the robotic problem have been addressed both analytically and with improved controller designs in numerous papers during the last decade. Indeed, contact instability is still considered one of the holy grails in the field of robotics. I will briefly review this literature here before reviewing the literature in the field of haptic interface design itself. First those papers from the robotics literature concerned with control design, then those that deal with the problem of contact instability with experimental investigations, and finally those containing analytical treatments will be reviewed.

Hogan has discussed the application impedance control [46] for the stable execution of contact tasts. stiffness control [], passive and active damping controls.

Xu, Hollerbach, and Ma present a nonlinear PD controller for contact transition in [107] Their controller features a set of PD gains which are a function of the force error and force error rate. During periods of robot motion away from the target, gains are increased with the aim of suppressing chatter.

Lin and Yae [60] present an improved impedance control design where contact force is extracted from a force sensor signal. A unified controller results, with no switching terms. The force feedback signal simply kicks in upon feedback of a non-zero force signal.

Mills [67], Mills and Lockhorst, [69] and Mills and Goldenberg [68] have presented a suite of discontinuous controllers for contact transition control along with stability proofs to guarantee asymptotic tracking of the commanded force upon contact, even given inadvertant bouncing. These papers call upon some of the russian litterature on discontinuous control [88], [87], [1].

Hyde and Cutkosky [50] conducted a comparative experimental study into the performance of five control designs which had appeared in the literature, each aimed at executing smooth, transition between motion and force control. The five controllers were simple discontinuous control, impedance control, active impact damping, active nonlinear damping, and input preshaping. Input preshaping was in fact a new introduction to the available contact control schemes. Noise sensitivity and ease in parameter selection and performance was compared.

Teleoperation

Hannaford and Anderson [41] discuss an experimental and simulation study into hard contact through a bilateral teleoperator. Forces sensed at the slave site are displayed at the master and positions sensed at the master are fed forward to the slave. A sixth order nonlinear but time invariant model was used for the human in the simulated system. The simulated system demonstrated behaviors very similar to the experimental setup. The effects of a heavier grasp on the handle were noted in experiment and duplicated in simulation with increased damping in the human operator model. Hannaford and Anderson mention on-line estimation of the parameters of the human operator as a possible extension for more robust control.

Robotics Literature: Analysis

These are controllers designed to execute tracking or minimize bouncing, in a discrete controlled robot. None of them address the destabilizing effects of the ZOH or intersample threshold crossing. These authors are primarilty interested in robot behavior, not emulation or exhibition of the dynamics of changing kinematic constraints. Controllers explore many methods to attain transition in minumum time (including adding physical compliance, virtual damping, switching laws, and so on), motivated in part by the extremely robust performance in such tasks which humans can demonstrate.

Our aims in coming up with virtual wall controllers are somewhat different than those of the robotics community interested in contact transition control. We actually want to keep the bouncey behavior, insofar as it represents the dynamics of the referent wall. But we want to extinguish the bouncey behavior which arises from the wall implementation in a haptic display device with a discrete controller. To the extent that destabilizing the effects shared, however, we are interested in the same issues as the robotics community. We are both interested in the dynamics of a discontinuous system, where making and breaking contacts is at play.

A number of papers in the spirit of identifying destabiling effects and designing controllers which compensate for those effects have appeared. For example, Eppinger and Seering treat the destabilizing effects of sensor/actuator non-collocation in force-controled robots in [29]. Effects of the robot and workpiece dynamics on the stability of a simple force-controlled system are considered when these dynamics intervene between sensor the point of application of control effort. Unstable behavior on the part of a robot is predicted using continuous-domain lumped-parameter models. Although dicontinuities were not analyzed within this paper, this unstable behavior often exhibits itself as a limit cycle of repeated contact and loss of contact with a workpiece. In the present chapter, by contrast, we have chosen not to treat the destabilizing effects of non-collocation. We assume that system dynamics do not intervene between the human/manipulandum contact and the sensors.

Regarding the half sample delay introduced by the zero order hold, the field of robotics has certainly acknowledged its destabilizing effect, and some robotic controllers have enjoyed the application of digital control design techniques which account for this delay. Neuman, for example, has contructed a fully discrete model of a robot manipulator [77]. Interestingly, though, as computer speeds increase and half sample delays grow shorter, attention to the benefits of design with digital techniques has waned. Most analyses and design efforts use continuous-domain methods again these days. Growing interest in the contact instability problem in robotics seems to have coincided with waning attention to sampled-data effects. Therefore, very little effort has been applied to analyze the effects of sampling on contact instability. I have not seen the sampling operator or zero order hold singled out for treatment in either design efforts or analysis efforts with regard to its effect on contact instability in the robotics literature. The effects of intersample threshold crossing have not been addressed in the design or analysis of robotic controllers to date.

Bartolini? [8] Utkin [96]

5.2.2 Haptic Interface Design Literature

In Chapter 4 of his thesis [86], Rosenberg presents the results of a human subject study on the rigid wall percept. The standard virtual wall algorithm, Eq. 5.2 was presented to human subects along with many variants (which incorporated exponential springs, thresholded dampers, unidirectional dampers, and position-offset dampers) for subjective ratings of "hardness", "crispness" (initial contact), "cleanness" (final release) and overall "wallness". Rather than defining these terms in physical variables, Rosenberg had chosen them to allow his subjects to fully characterize and also somewhat decompose the manifest behaviors of the various virtual wall algorithms. Among these behaviors was of course the same non-passive behavior underlying chatter, which Rosenberg describes as 'bouncy' contact. 'Sticky' release was also encountered in Rosenberg's damped walls. The various dampers were specifically designed to quell this undesirable and un-wall-like behavior. Rosenberg advocates a design methodology which he calls *design for perception*, in which one attempts to create percepts pertaining to virtual objects through perceptual decomposition rather than physical modeling. Perceptual decomposition involves human subject testing of various algorithms, some physically inspired, others simply tricks, to ascertain the optimum algorithm.

Although the *design for perception* approach may yield initial promise and point to some efficient shortcuts in algorithm design, we believe that perceptual decomposition is actually a more difficult problem than physical modeling. See Gillespie [34] for further discussion on this interesting topic. There exist many unanswered fundamental questions in psychophysics, and trial-and-error approaches become less attractive once the initial hurdles are overcome. The work in this thesis lies in the area of physical modeling, but certainly approaches like *design for perception* (where the *percept* rather than the algorithm take center stage) are useful to gauge the severity or consideration-worthiness of a problem. We take the work pertaining to the virtual wall of Rosenberg as further motivation for our own work. The supposition that contact instability is indeed a problem worth addressing with new controller designs is underlined by works like that of Rosenberg.

One of the earliest analytical treatments of contact instability associated with the virtual wall was in a paper by Minsky and Ouh-Young *et al.* [71]. This paper will be further reviewed in the next chapter —here I will just point out that the destabilizing effects of delay were addressed analytically. Interestingly, these authors attributed the delay to computational delays rather than the zero order hold operator. Virtual wall designs other than the standard controller were not explored, and design in the digital domain was not undertaken, though mention of digital analysis was made.

Colgate *et al.* present the results of an analysis of the passivity of certain virtual wall implementations to the virtual reality community in [23]. The analysis itself is covered in a pair of journal articles to be discussed in detail in the next chapter. The goals underlying this paper (and the supporting papers) are very much the same as the goals of [71] by Minsky *et al.*: "to delineate regions in parameter space that lead to suitable wall implementations". Colgate, however, formulates the problem as a question of passivity rather than one of stability.

Colgate endorses the use of a passivity criterion to characterize virtual walls rather than stability because passivity is a property of the wall alone, non-inclusive of the unpredictable human properties, making it possible to express passivity criteria without reference to properties of the human operator. Furthermore, the observed sustained or growing oscillations can be taken to be evidence of *active* walls since the human cannot be the source of energy as discussed above (because these oscillations are outside the range of voluntary motion, and sustained oscillations are not observed with physical walls). The main interpretation of the results of Colgate's passivity analysis is that an implementation of a virtual wall must contain some inherent physical damping if it is to be behave passively.

Colgate *et al.* cite the zero order hold as the path of energy introduction into virtual walls, and give an intuitive explanation similar the one above in section 5.1. Colgate *et al.* point out that, although coupled stability and isolated stability imply contact stability for continuous-time systems, these conclusions cannot be drawn for discrete time systems. Intersample threshold crossing and attendant simulation errors may emerge in sampled data systems. In a discrete analysis, Tsai and Colgate [95] treat the unilateral nonlinearity explicitly, but do not treat intersample effects.

Colgate's passivity analysis may be viewed as a very thorough and elegant treatment of the effects of the zero-order hold in this sampled data system. The elegance lies in the manner in which the specific dynamics of the human operator are excluded from the analysis and the fact that the end result may be applied to controller design. Colgate's contributions, however, lie in the area of analysis rather than design. New controllers or controller design methods which directly account for the destabilizing effects of the zero order hold are not suggested.

5.2.3 Simulation Literature

Numerous researchers in the field of numerical simulation are concerned with simulation across discontinuities, especially discontinuities embodied by changing kinematic constraints. Now that numerical simulation of multi-body dynamical systems is finding so much application in computer graphics, certain papers have appeared which address the problems directly. The desire to run these simulations in real-time has grown strong of late with emerging interest in *interactive* systems, bringing this literature close in spirit to our concerns.

Researchers in this field, however, have the luxury of being able to neglect the dynamics of the human in the consideration of accurate simulation across discontinuities since there is no loop closed through mechanical variables when the human is coupled through a unmotorized interface device. Furthermore, the effects of instability using only visual display are far less disturbing than in the case of haptically displayed instability. Thus less attention has been paid to difficulties in simulating across discontinuities in this field to date. Non-physical behavior due to limited update-rate is often dismissed as less important since it can be effectively treated with more computing power. Such treatment is less readily applied in haptic interface because of hardware interfacing requirements.

Howe [48] presents a technique which essentially amounts to half-sample prediction to account for the equivalent half-sample delay introduced by a zero-order hold in real-time flight simulation with motion display. Howe also prevents the effects of computational delay from surfacing with similar simulate-ahead strategies. Given that motion display does not actually close the loop through mechanical variables as does haptic display (unless feed-through dynamics are present), Howe does not need to include human dynamics in the model which is used for half-sample prediction.

A paper by Lin and Howe [59] addresses the issues involved in real-time simulation of a discontinuous system with a discrete constant step-size simulation algorithm. Their approach to the real-time simulation of systems with discontinuities takes care of errors arising from an occurrence of a discontinuity between simulation steps. The dynamic equations of a control subsystem with discontinuities are integrated off-line with a sufficiently small step-size, repeated over a matrix of initial conditions and inputs, to produce a function table which may be used during run time to efficiently account for intersample placement of a discontinuity.

In summary, although applied in robotics, it appears that digital control design techniques have not been applied in the field of haptic interface to date. While appearing to some extent in numerical simulation, compensation for the effects of intersample threshold crossing in simulation across discontinuities has not been applied in haptic display. The application of deadbeat control to correct for errors of intersample threshold crossing in a discontinuous system is new.

Our approach to the problem of contact instability in virtual walls can be contrasted with that of most other researchers to date. Rather than setting out to delineate regions in parameter space which will ensure passivity of a virtual wall using a particular (somewhat standardized) controller [25] [23] [24], we embarked on another effort —to design an altogether new controller which would meet some special performance criteria. These were in fact a rather stringent set of performance criteria: that the controlled system (despite its sampled data structure) behave exactly as another continuous but switching system, as discussed above. Under the virtual wall application, the criteria consisted of non-introduction of energy into the system upon contacting the wall.

Although methods for the design of sampled data controllers have appeared widely in the literature, and methods for handling switching models in constant step-size simulation algorithms also exist [48], application of both methods to sampled data controller design has apparently not been made. The switching sampled-data controllers developed in this chapter do not have precedent in the literature. Although contact instability has received much research attention in the Robotics literature, authors have not ascribed this instability to the sampled-data implementation of robot controllers.

5.3 Modeling the sampled data system

To expound the controller designs, I discuss another discontinuous system, simpler than, but still representative of the virtual wall: the lossless bouncing ball. Shortly I will defend the bouncing ball as a suitable model of human exploration of a virtual wall through a haptic interface, but first, I discuss our use of the bouncing ball model as a kind of work bench for the design of virtual wall controllers.

Basically, we seek a simulation algorithm for the elastic floor upon which a ball bounces which yields 'realistic' bouncing behavior, yet which adheres to certain 'structural restrictions' placed on the algorithm itself. We know that a lossless ball bouncing on a perfectly elastic floor should bounce forever, never gaining or losing height. Any simulated behavior in which the ball bounces higher, lower, or even irregularly shall be deemed 'non-physical'. For a ball with damping, any simulated ball motion which deviates from the motion of its referent (continuous) damped bouncing ball will indicate problems, or a failed floor simulator design. Note that the motion of the continuous bouncing ball is easily produced with simple simulations, or even with the careful use of two switching analytical solutions.

The structure of the simulator itself will be further elucidated below, but stated simply, the simulator structure is modeled after the very sampled data system which is the haptic interface displaying a virtual wall. Thus, upon coming up with a suitable floor simulation, that simulation algorithm may be reinterpreted as a wall controller and implemented directly in the actual physical hardware. So the immediate goal which directs the controller design is simply to eradicate 'non-physical' simulated behavior which arises in the bouncing ball simulation because of the 'structural restrictions' placed on the simulator.

5.3.1 The Bouncing Ball as Allegory for the Virtual Wall

The bouncing ball, I shall now argue, is in fact a rather good model of a human interacting with a virtual wall through a haptic interface. Again, the tendency of the manipulandum to chatter or iteratively bounce against a virtual wall is the behavior which we aim to study. Presumably, if the bouncing ball is a good model, its unstable *simulated* behavior using a certain floor simulation algorithm will predict chattery *controlled* behavior when that 'floor simulator' is implemented as a 'wall controller'. In order to generate a particularly simple bouncing ball model (and likewise a simple work bench), we model the ball and floor without any dissipative elements. We work under the premise that bouncing ball instability or unbounded growth (due to the introduction of energy from an unsuitable simulation algorithm) will indicate unfitness of the corresponding wall controller (where the introduced energy will possibly only cause sustained oscillations because of damping inherent in the physical system).

Figure 5.6 shows a massive ball B in two configurations. The configuration in Figure 5.6 a) corresponds to the *floor off* condition. Here, the ball is being acted on solely by the force of gravity. In Figure 5.6 b), corresponding to the *floor on* condition, the ball is being acted on by the force of gravity and the force of a special spring $k(y,t)$. The rest position of the wall spring is taken to be zero ($y_{wall} = 0$).

The ball represents the manipulandum and the hand or finger of the human operator. For now, elastic and dissipative effects in the human and manipulandum are not modeled. The force of gravity represents a constant force exerted by the human on the manipulandum. Backup for such a crude

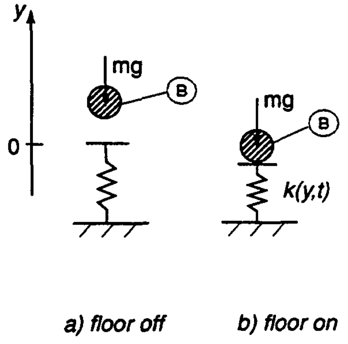


Figure 5.6: *Modeling the Bouncing Ball*

approximation of the human in the virtual wall system under study is provided, as mentioned in section 5.1, by noting that the observed chatter is much higher than the frequencies characterizing the intentions of the human and that a human does not need to do anything once oscillations begin in order to sustain them except passively maintain that hand impedance found to be destabilizing. Various gravity fields will be used to represent various forces exerted by the human. A representative impedance (inertial, damping, and spring forces) for the human is also neglected for the present for simplicity.

The spring force of the floor depicts the virtual wall, but, as mentioned above, in order to allow for the subsequent reinterpretation of the 'floor' as a controller in a sampled-data system, the floor simulator is specially structured as follows. Rather than by Newton's impact law (with a coefficient of restitution), the floor is modeled as a compressible spring with a constituent law f=k(y,t) to be determined by the designer. Simulations of this model are allowed to communicate with simulations of the ball only at certain time points (the sampling times). Furthermore, the force response of the floor shall be held constant between sampling times to depict the zero-order-hold. The floor is thus simulated as a discrete system and the ball as a continuous system. While the ball depicts the manipulandum and human, (plant) the floor depicts the discrete controller.

Figure 5.7 shows this discrete floor as the feedback controller in a block diagram with the ball as plant. The discrete floor controller $C(z)$ is shunted into the loop only during the *floor on* periods of simulation by the unilateral nonlinearity.

Our simulations thereby include the zero order held forces and appropriately compute the response of the continuously modeled ball to these discontinuous (staircase-shaped) forces. The switching times are also constrained in this simulation as they are in the sampled data system, to lie on

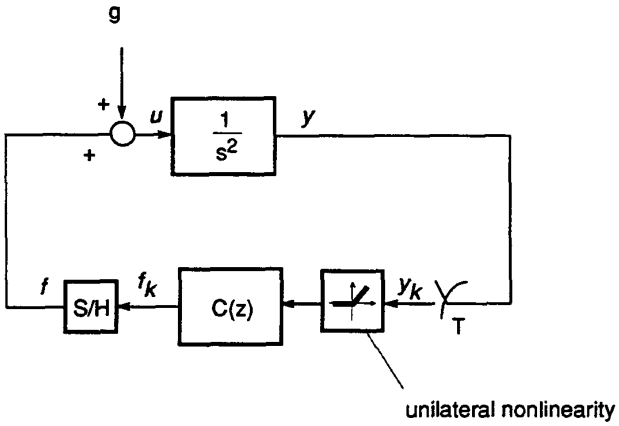


Figure 5.7: *Sampled Data System-inspired Block Diagram for the Bouncing Ball Simulator*

the sampling times.

We begin with a floor control law for simulation like that most commonly used in virtual walls (the control law inspired by its continuous time counterpart), namely $f_{spring} = k(y_k - y_{floor})$. For the floor, we set $y_{floor} = 0$ as in Figure 5.6. The differential equation (model) for a unit mass responding to gravity and the force of a spring is simply:

$$\ddot{y} = -g - f_{spring} \tag{5.3}$$

which has the following equivalent form as state-space model, using $x = [\ y \quad \dot{y}\ ]'$:

$$\dot{x} = Ax + b(-g - f_{spring}) \tag{5.4}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \text{and} \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Note that the reaction force of the spring is a forcing term (on the right-hand side) in this model. The motion of the ball is simulated in intervals, each the length of one sampling period T, with an ODE solver. At each sampling point (between intervals) an indicator function is checked (whether the ball is inside the domain of the floor). If the indicator function evaluates true, the reaction force of the spring, f_{spring}, is computed according to the control law, and held constant for the duration of the next sampling period. If the indicator function evaluates false, f_{spring} is set to zero.

Pseudocode for our simulator is given in Table 5.2. Appendix A contains MATLAB code for this algorithm.

Table 5.2: **Pseudocode for the Sampled Data Bouncing Ball**

```
$k = 0$
loop {
    apply ODE solver to (5.3) from $t = kT$ to $t = (k+1)T$
    append interval to stored solution vector $x(t)$
    if $(y < 0)$ then $f_{spring} = K * y$
    else $f_{spring} = 0$
    $k = k + 1$
}
plot $x(t)$.
```

In order to produce the motion of a continuous bouncing ball for comparison, several methods can be used, (including simple evaluation of model solutions). The method most parallel in structure to the above uses an ODE solver on time-intervals which are pre-determined from the solution. The following will demonstrate determination of the switching times.

Starting from a state outside the floor, the time to floor strike is computed using the model of a free unit mass flying ball:

$$\ddot{y} = -g \tag{5.5}$$

The solution to this simple model is of course:

$$y(t) = y_0 + v_0 t - \frac{1}{2} g t^2 \tag{5.6}$$

The time to floor strike is simply its root, given by:

$$\Delta t_I = -v_0 - \sqrt{v_0^2 + 2gy_0} \tag{5.7}$$

where y_0 and v_0 are the initial state of the free-flying ball.

The *floor off* model is used to simulate the motion for the time period Δt_I. From the time of wall entry (on threshold), the time to exit the floor is computed using the solution of the *floor on* model given the floor entry state. The *floor on* model is simply a sprung mass (this time with the spring incorporated into the left hand side of the equation):

$$m\ddot{y} + ky = -g \tag{5.8}$$

which has an equivalent from as a state-space model:

$$\dot{x} = Ax + b(-g) \tag{5.9}$$

where

$$A = \begin{bmatrix} 0 & 1 \\ -k/m & 0 \end{bmatrix} \quad \text{and} \quad b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

The solution to this differential equation is:

$$x(t) = C_1 cos(\omega t) + C_2 sin(\omega t) - g/w_0^2 \tag{5.10}$$

where

$$\begin{aligned} C_1 &= y_0 + g/w_0^2 \\ C_2 &= v_0/w_0 \\ w_0 &= \sqrt{k/m} \end{aligned}$$

The first root of this equation, or time Δt_{II} which yields $y = 0$ is:

$$\Delta t_{II} = \frac{1}{w_0}\left[\pi + atan2(C_1, -C_2) + sin^{-1}\left(\frac{g/w_0^2}{\sqrt{C_1^2 + C_2^2}}\right)\right] \tag{5.11}$$

where $atan2$ is the four-quadrant arc tangent function.

The *floor on* model is then used in the ODE solver for the time period Δt_{II}. Then the the process starts over.

Pseudocode for this algorithm is given in Table 5.3. Appendix A contains MATLAB code for the same algorithm.

Figure 5.8 shows the simulation results using the "Sampled Data" simulator (Table 5.2) along with the results of the continuous bouncing ball simulator (Table 5.3) for reference, which is shown with a dashed line. The staircase-shaped trace of the reaction force from a unit-stiffness spring,

Table 5.3: **Pseudocode for the Continuous Bouncing Ball**

```
loop {
    compute time to floor strike $\Delta t_I$ using (5.7)
    apply ODE solver to (5.5) from $t$ to $t + \Delta t_I$
    append interval to stored solution vector $x(t)$
    compute time to floor exit $\Delta t_{II}$ using (5.11)
    apply ODE solver to (5.8) from $t$ to $t + \Delta t_{II}$
    append interval to stored solution vector $x(t)$
}
plot $x(t)$.
```

f_{spring} is also ploted.

Figure 5.9 shows a close-up of the boxed portion of Figure 5.8.

We very quickly note that this floor produces non-physical behavior; the ball bounces higher and higher. At this point, the design of improved controllers is underway.

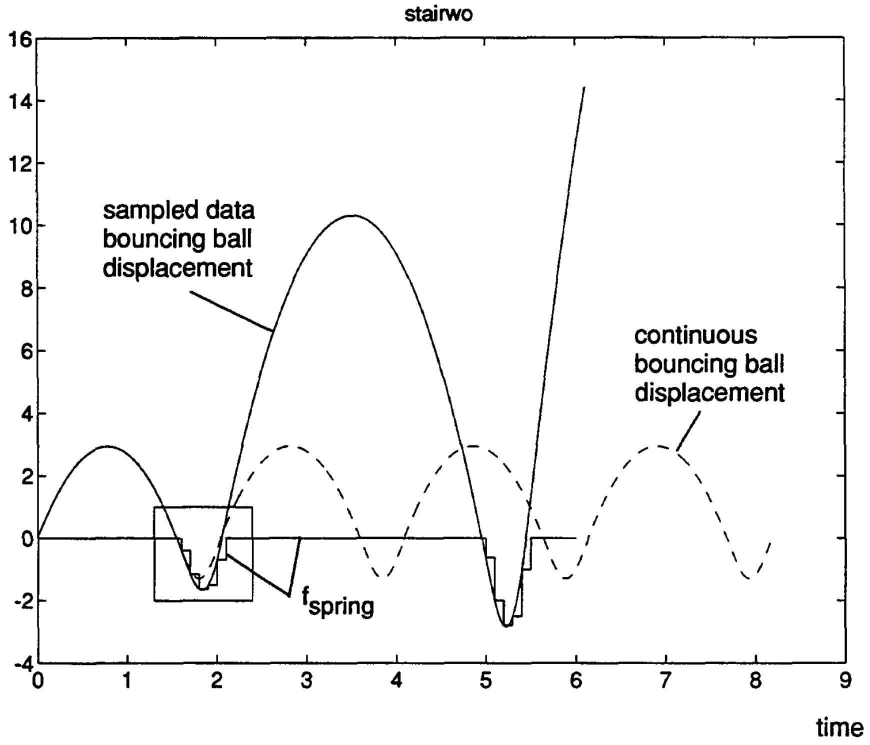


Figure 5.8: *Sampled Data Algorithm Simulation Results*

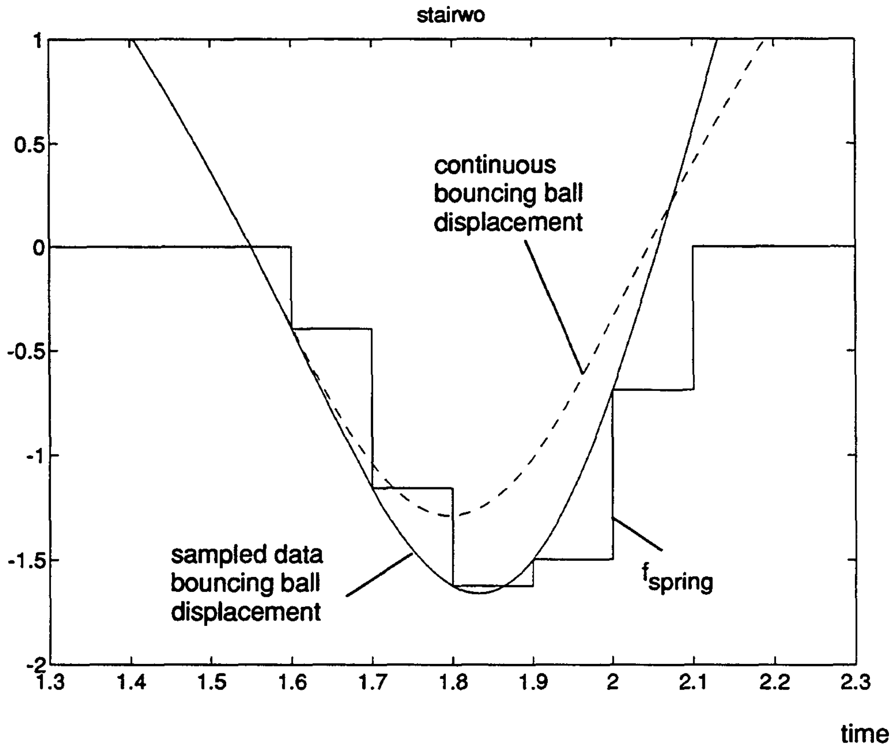


Figure 5.9: *Sampled Data Algorithm, Close-up*

5.4 Controller design

Two controller designs will be presented, the first based on half-sample prediction and the second on design in the digital domain. Both of these controllers are intended to compensate for the destabilizing effects of the zero order hold within our sampled data system. An enhancement to both of these controllers, which compensates for the effects of intersample threshold crossing, will be presented in the third part of this section.

5.4.1 Half Sample Prediction Controller

Our first improved controller is inspired by noting that the effect of the zero order hold can be approximated by a half-sample time delay. An improved controller will be constructed by adding half-sample prediction to the algorithm in the hopes of cancelling the effect of the zero order hold.

We already have a model of the target system (a sprung mass) in hand in the form of Eq 5.8. At each sample time $t = kT$, we can simulate ahead using this model or, even easier, evaluate its solution, Eq. 5.10, at $t = kT + T/2$ and starting from the present state $[y_0 v_0] = x(kT)$, to predict the ball's position a half sample ahead. This predicted position is then used in the standard control law (Eq 5.2).

Note that the model whose response is the target (the sprung mass, Eq. 5.8), rather than the model of the mass responding to the zero-order held force (the mass on wall, Eq. 5.4) is used to make the predictions.

Our simulation pseudocode using this half-sample prediction now looks like the following:

Figure 5.10 shows the simulation results of the above half-sample prediction pseudocoded algorithm. The spring stiffness is unity $K = 1$.

Figure 5.11 shows a close-up of the boxed area in Figure 5.10. Here one sees how the trace of the zero-order held spring force intersects the continuous floor position trace approximately midway between sample times. This propitious intersection leaves half of the inscribed area above and half below in contrast to the staircase plot which the reader may recall from Figure 5.3. Our new algorithm does not climb "uphill both ways" as that of Figure 5.3.

The other interesting thing to note in Figure 5.11 is that, for the last wall-on sample period, the wall is actually exerting a tensile force, pulling on the operator. During this period, since the manipulandum is moving in the direction away from the wall, the operator is doing work on the wall. During motion away from maximum wall penetration, the wall is for the most part returning work to the operator, except for this last sample period. This fact will become important and will be further discussed in the analyses of the next chapter.

Though the half-sample prediction method outlined above yields vastly improved results over

Table 5.4: **Pseudocode for the Half Sample Prediction Bouncing Ball**

```
$k = 0$
loop {
    apply ODE solver to 5.3 from $t = kT$ to $t = (k+1)T$
    append interval to stored solution vector $x(t)$
    if $(y < 0)$ then {
        $A = (y0 + g/w_0^2)$; $B = v_0/w_0$;
        $y_{predict} = Acos(t + T/2) + Bsin(t + T/2) - g/w_0^2$
        $f_{spring} = K * y_{predict}$
        }
    else $f_{spring} = 0$
    $k = k + 1$
}
```

those given by algorithm 5.2, the results are still not perfect. Deviations from the desired bouncing path can already be seen in Figures 5.10 and 5.11, but excursions become especially apparent if the algorithm is allowed to continue for some time, as in Figure 5.12. Here we see that the bouncing height is irregular, sometimes higher, sometimes lower than the target height.

Reasons for this erratic behavior are twofold. Firstly, the ZOH is only approximated by a T/2 sample delay; its full effect is more complex. See, for example [31]. The next section will present a design using digital controller design tools which fully accounts for the effects of the ZOH.

Secondly, the wrong control law will be used to compute the reaction force for certain portions of those sample periods which contain the entry and exit, thus exerting a force inappropriate to that portion of the time period. Stated another way, turning on and turning off of the floor control law (entry into *if* block of pseudocode) will not necessarily occur when the ball height is $y = 0$. A fix for this second phenomenon which we call intersample threshold crossing will ~~be presented~~ in the section following the next.

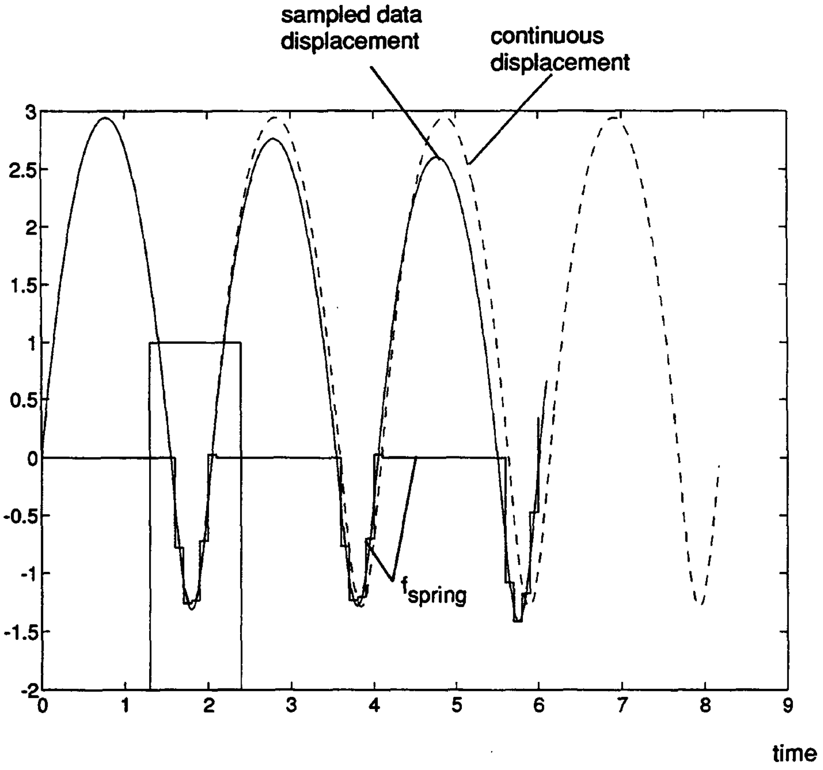


Figure 5.10: *Half Sample Prediction*

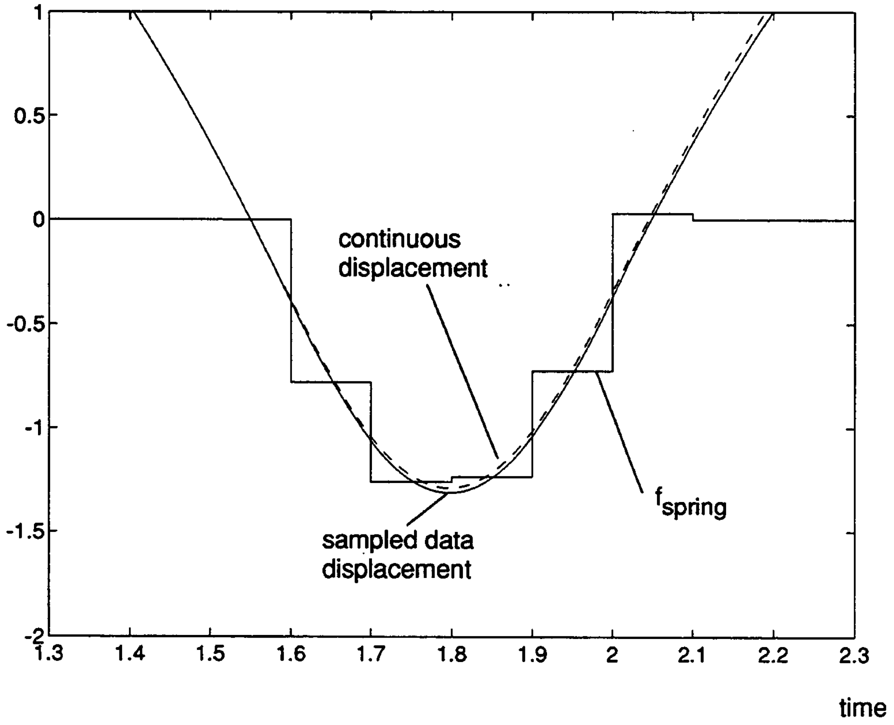


Figure 5.11: *Half Sample Prediction, Close-up*

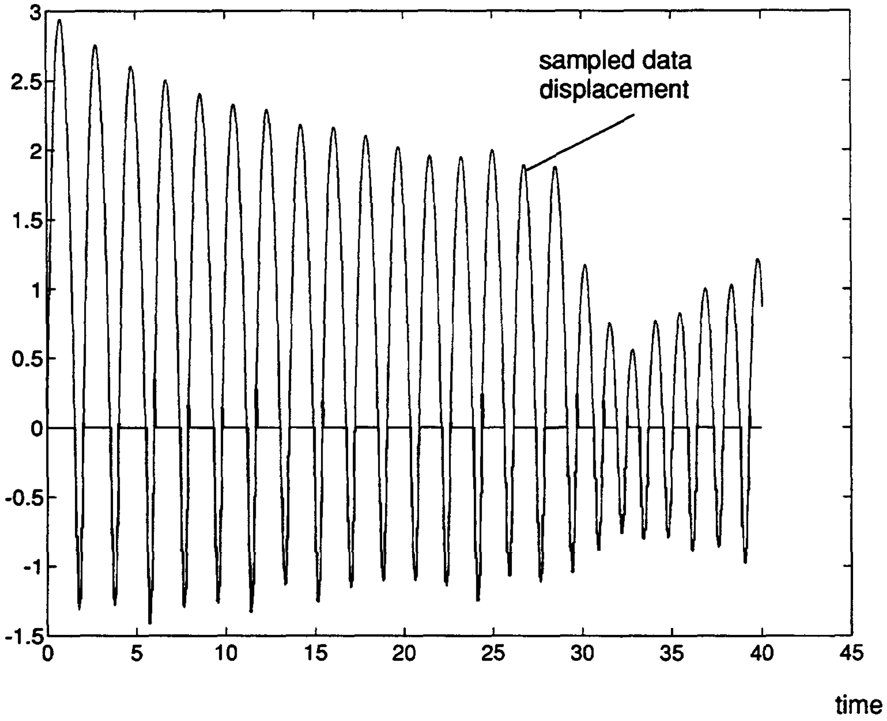


Figure 5.12: *Half Sample Prediction, Long-term Simulation*

5.4.2 Design in the digital domain

Another approch to the design of a controller for our bouncing ball simulator (an in turn for the haptic display of a virtual wall) exists. It is controller design in the digital domain. Our interim goal is to design a controller for a discretized plant, such that the response of the closed loop discrete system is the same as the desired (continuous bouncing ball) response on the sampling times.

First the ZOH discrete equivalent of the desired dynamics (a sprung mass) are found using a table of $\mathcal{Z}$ transforms.

$$\mathcal{ZOH}\left\{\frac{1}{s^2+\omega_0^2}\right\} = \frac{(1/\omega_0)(1-cos\omega_0 T)(1+z)}{z^2-(2cos\omega_0 T)z+1} \tag{5.12}$$

Where $\mathcal{ZOH}\{\cdot\}$ denotes the zero-order hold discrete equivalent of the bracketed expression. Note that $\mathcal{ZOH}\{\cdot\} = (1-z^{-1})\mathcal{Z}\{\cdot\}$.

This discrete equivalent has two complex poles and one zero. The pole locations (roots of the characteristic equation), λ_1 and λ_2, as shown in Figure 5.13, are identified as the desired root locations for the closed loop system comprising the controller being designed and the simple plant $H(s) = 1/s^2$. These root locations correspond to the response of the referent system, a sprung mass, expressed in the digital domain.

We will perform the design of that controller in the digital domain. For this purpose, the ZOH discrete equivalent of the plant $1/s^2$ is found.

$$\mathcal{ZOH}\left\{\frac{1}{s^2}\right\} = \frac{T^2(z+1)}{2(z-1)^2} \tag{5.13}$$

Using full state feedback control, the poles of this system may be placed arbitrarily. We simply choose to place the roots of this controlled system at the root locations λ_1 and λ_2 using pole placement. The full state feedback gains $k = [k_1 k_2]$ which place the closed loop dynamics of the sytem at these target locations are available from a pole placement algorithm such as that known as Aizerman's method.

The response using this controller is essentially the same as that of the previous section, the half-sample prediction controller. Figure 5.14 shows the response of the sampled data bouncing ball with the new discrete controller overlaid on top of the target displacement trajectory. The system is still not perfect, due to errors in switching the controller at the wall threshold (intersample threshold crossing). The next section will address this very problem, producing results in the end from the sampled data controller which are indistinguishable from the continous target system.

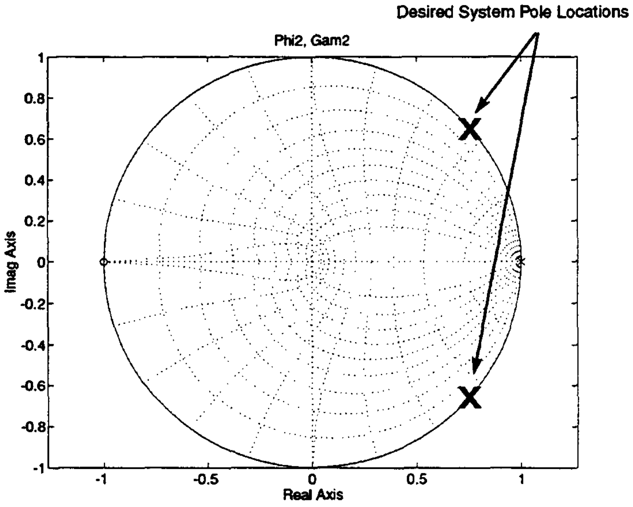


Figure 5.13: *Desired root locations on the unit circle*

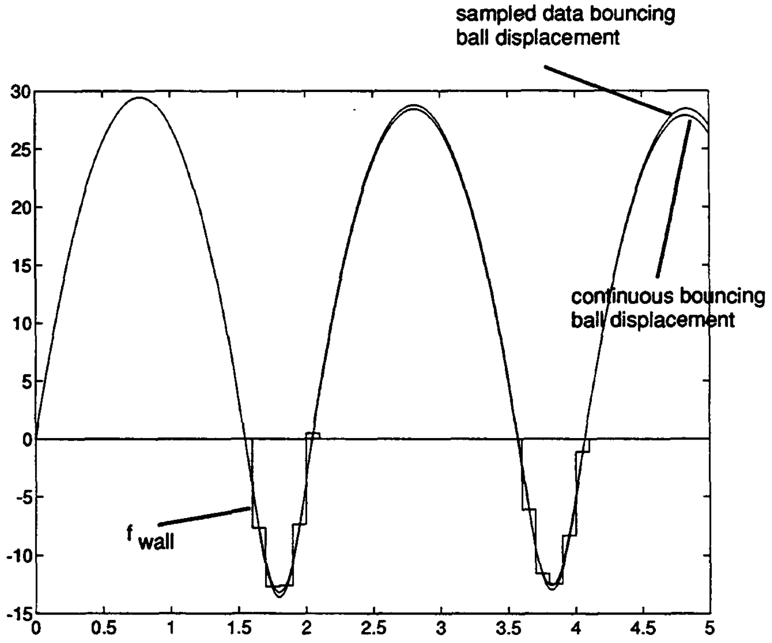


Figure 5.14: *Performance of Controller designed in the digital domain*

5.4.3 Compensation for Asynchrony

The effects of inter-sample threshold crossing can be fully accounted for by using solutions of the two models, the free-flying mass and the sprung mass. For clarity, the approach is first outlined: First, we deduce the inter-sample entry and exit times from information available at the sampling times. We then predict, again using the model solutions, what state the ball would be expected to attain, if the system were continuous, at the first sampling time outside of the floor. Finally, we drive it to that desired state (as shown below) with the last two (zero-order-held) force values inside the floor.

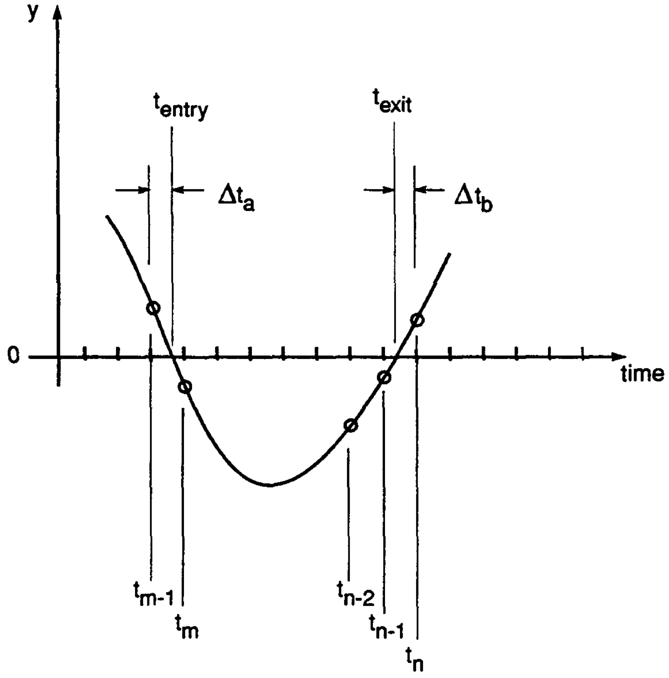


Figure 5.15: *Sampling Points in a Typical floor Strike*

Figure 5.15 shows the arbitrary placement of the sampling times on the motion path of a simulated strike of the floor. The first sampling time for which the ball is inside the floor is designated t_m and the first outside of the floor is designated t_n. Reference to these time points will be made in the following discussion.

The state of the ball at the first sampling time outside of the floor, $x(t_n)$ encapsulates the action of the floor simulator. The state of the ball at t_n resulting from a continuous floor is obviously not the same as the state resulting from either the typical algorithm floor (see Figure 5.9) or the improved half-sample prediction floor algorithm (see Figure 5.11). But, having assumed models for both the *floor on* and *floor off* conditions, the state of the ball resulting from a continuous floor (which we shall call x_d, (d for *desired*) can be extracted from the state information available on the sampling times. To find this state is a multi-step process as follows:

First, the root of the free-flying ball model is found, using the known state at the last sampling time before entry $x(t_{m-1})$ in Eq. 5.7. This time interval is designated Δt_a (see Figure 5.15).

The full state at floor entry $x(t_{entry})$ is found using the solution to the free-flying ball model, Eq. 5.6 and its derivative,

$$\dot{y}(t) = v_0 - gt \tag{5.14}$$

by evaluating these at $t = \Delta t_a$ and $[\; y_0 \quad v_0 \;]' = x(t_{m-1})$.

Now the time at which the exit from the floor is made, t_{exit}, is found using the initial condition $[\; y_0 \quad v_0 \;]' = x(t_{entry})$ in the root of the sprung mass solution (Eq. 5.11),

Note that the state at t_{exit} is already known, quite simply, because it is an undamped wall:

$$x_{exit} = \begin{bmatrix} y_{entry} \\ -\dot{y}_{entry} \end{bmatrix} \tag{5.15}$$

where the shorthand x_{entry} stands for $x(t_{entry})$.

Knowing the time of exit t_{exit}, the time remaining to the first sampling period outside of the floor Δt_b (see Figure 5.15) is available:

$$\Delta t_b = t_n - t_{exit} \tag{5.16}$$

where $t_n = kT$ and k is the smallest integer such that $y(kT) > 0$ since the last floor encounter.

Finally the desired state at time t_n can be computed by evaluating the solution to the free flying mass model, Eq. 5.6, starting at the exit state $[y_0 v_0]' = x_{exit}$ and $t = \Delta t_b$. This state $x(t_n)$ we shall call the desired state x_d.

Driving the system to the desired state

We now know where this system is to be driven if it is to behave as the continuous bouncing ball. Now the problem has become how to shape the control force f so that, at t_n, the state will arrive at x_d.

This is a problem of controller design in the digital domain. We start by discretizing the continuous part of the system, which is of course only seen by the discrete controller at the sampling times kT.

We find the ball's zero-order hold equivalent so that the effects of the zero-order hold is included in its discrete representation. Since our model of the display device and human are quite simple, their zero-order hold discrete equivalents are easily found by an application of the definitions of Φ and Γ (See, for example: [31]):

$$\Phi(T) = I + AT + \frac{(AT)^2}{2!} + ... = exp(AT) \tag{5.17}$$

$$\Gamma = \left[\int_0^h \left(I + \tau A + \frac{\tau^2 A^2}{2!} + ...\right) d\tau\right] b \tag{5.18}$$

$$= \left[T + \frac{T^2 A}{2!} + \frac{T^3 A^2}{3!} + ...\right] b \tag{5.19}$$

But for our model Eq. 5.3, A^2 is a matrix of zeros, so in terms of the sampling period T, Φ and Γ are simply:

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \Gamma = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} \tag{5.20}$$

Given that the discretized system is only second order and that it is fully controllable, it will only take two steps to drive it to any desired state.

The response of a discrete system $\{\ \Phi \quad \Gamma \quad c\ \}$ to the input sequence $u(kT)$, $k = 0, ...n$ can be expressed:

$$x(n) = \Phi^n x(0) + \mathcal{C} \begin{bmatrix} u_{n-1} \\ u_{n-2} \\ \vdots \\ u(0) \end{bmatrix} \tag{5.21}$$

Where the controllability matrix $\mathcal{C}$ is given by:

$$\mathcal{C} = \begin{bmatrix} b & \Phi b & ... & \Phi^{n-1} b \end{bmatrix} \tag{5.22}$$

Given controllability ($det(\mathcal{C}) \neq 0$), this equation can be inverted for the control sequence

$$\begin{bmatrix} u_{n-1} \\ u_{n-2} \\ \dots \\ u(0) \end{bmatrix} = \mathcal{C}^{-1}(x_d - \Phi^n x(0)) \tag{5.23}$$

Since our system is only second order, it will only take two inputs to drive the system to state x_d:

$$\begin{bmatrix} u_{(n-2)} \\ u_{(n-1)} \end{bmatrix} = \mathcal{C}^{-1}(x_d - \Phi^2 x0) \tag{5.24}$$

where $x0$ is the state $x(t_{n-2})$, two samples before the first sample outside of floor, with n pertaining to Figure 5.15.

The control value u_1 is used at t_{n-2} and u_2 is used at t_{n-1}

Deadbeat Control results

Figure 5.16 shows the simulation results of the above T/2 prediction pseudocoded algorithm, 5.4.1. The spring stiffness is $K = 1$.

Figure 5.17 shows a close-up of the boxed area in Figure 5.16.

5.5 Experimental Results

The half-sample prediction controller described in the forgoing section was coded in C++ and tested experimentally using our haptic interface. To facilitate collection of intersample position, the control law was evaluated only every tenth servo cycle, while data was collected every servo cycle. Thus a wall controller was mimicked whose implementation had a sampling rate one-tenth that of the actual servo rate. A virtual wall was displayed in the center of the workspace of one key. Sequential comparisons of virtual walls rendered either with the standard controller or the new controller were made. Controllers were exchanged, sampling rate varied in increments of 50 Hz from 100 to 1000 Hz, and target stiffness was varied during run-time through a simple interface.

The new virtual wall controllers performed significantly better than the old. Walls rendered with the new controllers did not support sustained oscillations for those parameter values under which the old virtual walls in fact did support sustained oscillations.

Figure 5.18 shows the bouncing behavior of the old wall. Four strikes were made against the same wall, but with alternating controllers in action. Approximately the same posture and muscle activation levels were adopted by the human subject. The first and thrid strikes were against a

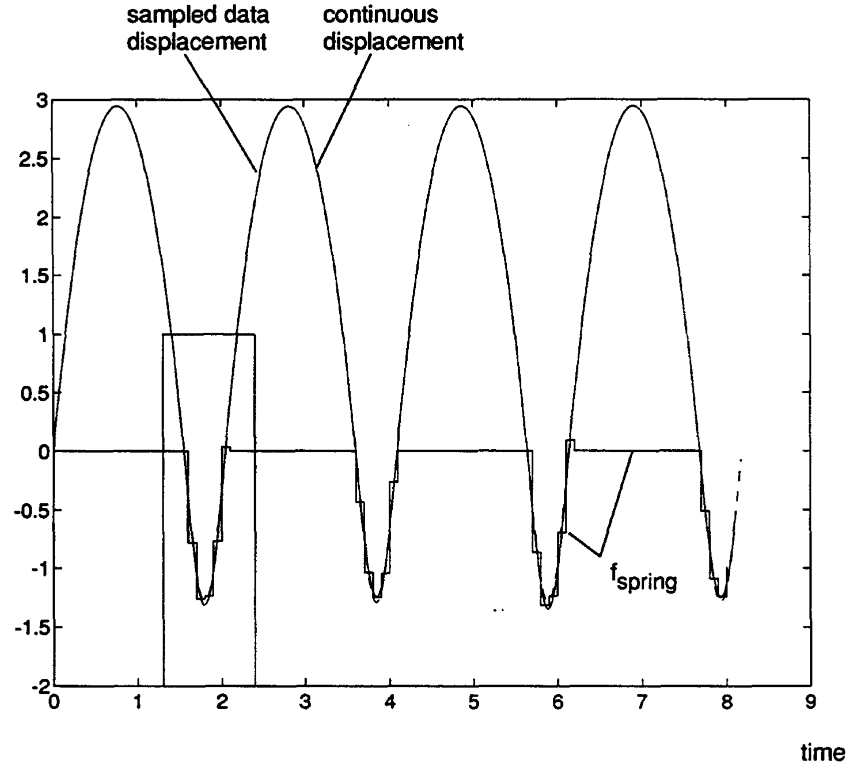


Figure 5.16: *Full Control Algorithm*

virtual wall using the old controller. The second and fourth strikes were against a wall using the new controller. The controller out-performs the old, as evidenced by the lack of chatter.

Figure 5.19 shows a closed -up of the boxed portion shown in Figure 5.18. Using a wall stiffness scaled to unity, the control effort (force for display) from both controllers is shown overlayed on the position trajectory of the manipulandum. The solid line shows the old control effort (displayed during this time period) and the dashed line shows the new control effort (not displayed during this time period). The intersection of the new control effort with the position at the half-sample times is evident.

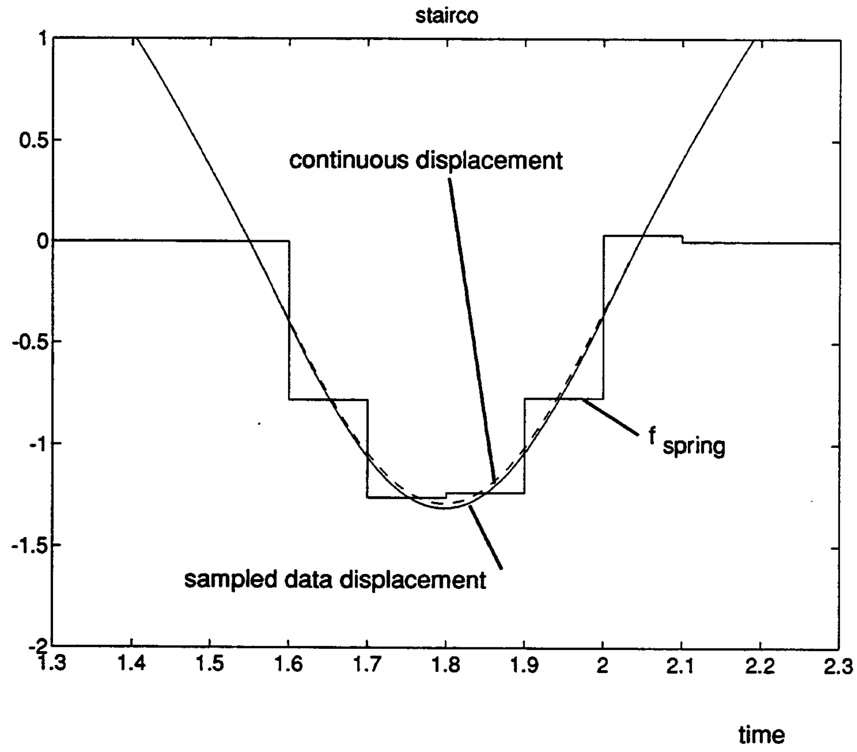


Figure 5.17: *Full Control Algorithm, Close-up*

5.6 Discussion

The forgoing controller exposition and experimental presentation have concentrated solely on the rendering of undamped virtual walls. New controllers were developed which prevent the chatter commonly observed in undamped walls from arising. Yet the new controllers use full state feedback, with gains on both sensed position and sensed velocity. Perhaps it is not fair to compare a wall control law which uses both position and velocity feedback gains with wall controllers which only use position feedback gains. Comparisons between the new controllers and *damped* virtual walls would be more fair. Damping is often added to virtual walls to enhance their stability, usually by trial and error. This trial and error process in the design of damped walls makes their direct comparison difficult. It can indeed be said, however, that the significant improvements exhibited by the new controllers can be attributed to the addition of positive velocity feedback.

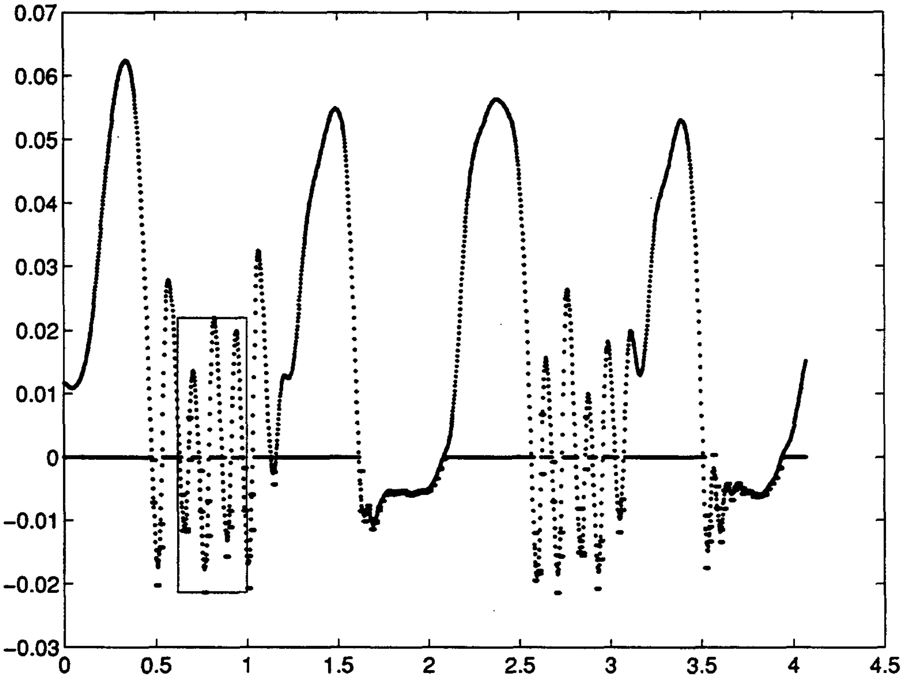


Figure 5.18: *Four stikes of a virtual wall, with two controllers*

The point of the new controllers, then, is not so much their highly improved performance as promulgated in the previous section, but rather that a method for designing feedback gains has been presented which, when implemented in the sampled data setting, will exhibit the desired stiffness. Damped walls with desired stiffness and damping can also be designed with the same method. The model to be used in the half-sample prediction controller would then be a damped second order oscillator rather than the undamped oscillator used in the exposition of this chapter. The desired pole locations in the design-in-the-digital-domain procedure would turn up inside the unit circle. The same Aizerman pole placement algorithm, however, would come up with appropriate feedback gains on position and velocity just as in the undamped case shown. The risk of coming up with a wall which feels damped because excessive damping was chosen (providing more stabilizing influence than necessary) is not a problem with these new design techniques.

The half-sample prediction controller features another important advantage over the old damped virtual wall design. That is, when a first-difference approximation is made for the velocity from a position signal, appropriate model-based filtering will automatically be added. Essentially, the

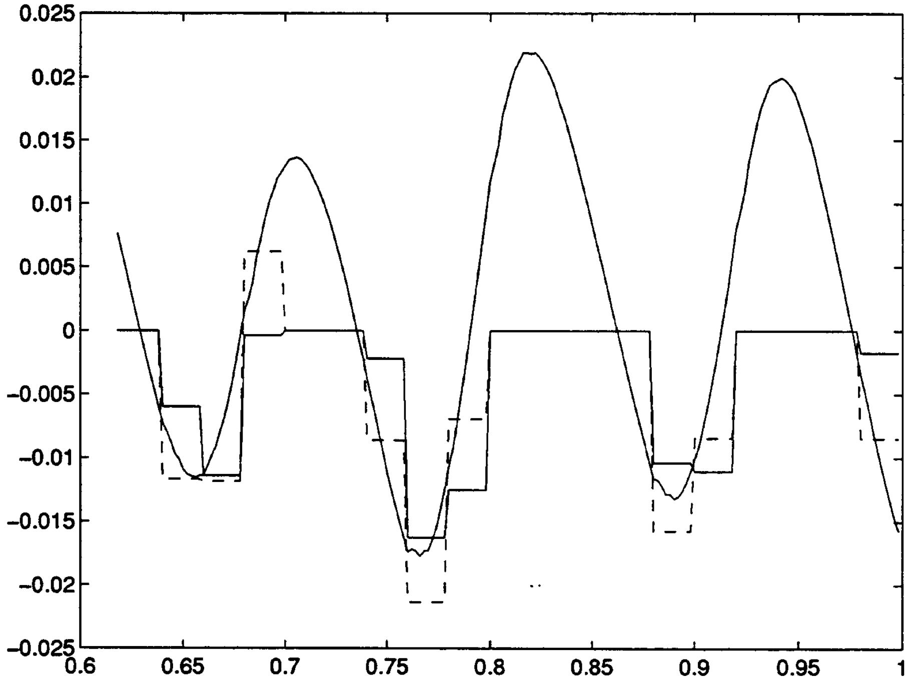


Figure 5.19: *Four stikes of a virtual wall, with two controllers*

use of the present sensed velocity in a model to produce a future state is like a Kalman filter. Many authors have observed and Colgate has shown that increased damping coefficient can lead to unstable behavior from the virtual wall when that damping coefficient is used on a first-difference approximated velocity.

These comments pertain to the compensation for the zero order hold. The dead-beat control techniques may of course also be used in the design of damped walls. This technique for the quelling of effects from intersample threshold crossing does not have precedent in the literature.

5.7 Summary

This chapter has addressed the formulation of controllers designed to create the illusion, for a human operator, of a passive wall which opposes motion with spring forces when the operator drives a manipulandum past a threshold. A set of virtual wall controllers has been presented which are immune to two certain destabilizing factors which otherwise play a large role in the implementation of virtual walls, causing chattery behavior. Both of these factors are a consequence of the sampled data setting within which virtual wall controllers must operate. First, the zero order hold effectively introduces loop delays which, under position feedback (as called for by a stiff wall) introduce energy into the closed loop system. The effects of the zero order hold are quelled in the new virtual wall controllers using two design techniques: half-sample prediction and design in the digital domain. Both of these controller enhancements were shown to substantially improve virtual wall performance (decrease chatter) in simulation and in experiment. Close scrutiny of simulation, however, shows that something is still amiss in these sampled-data virtual walls: the issue of intersample threshold crossing, the second factor which we address.

The controller for the virtual wall is a *switching* controller, which attempts to cause the manipulandum to take on the dynamics of a two-mode system —an object alternately in contact with a compliant wall and in free motion, as driven by the human operator. Because the on/off switching is based on a signal which is only discretely sampled, timing errors are introduced into the switching behavior of this controller. Turn-on and turn-off times are not synchronous with the wall encounter times, but rather with the next available sampling times, occurring quite independently of the threshold crossings. The effects of intersample threshold crossing may be fully counteracted, however, by model-based deduction of the timing errors from state information collected on the sample times and with an application of dead-beat control. The fact that the controller is discrete works to our advantage this time, since deadbeat control is able to perfectly compensate for the errors of intersample threshold crossing and deadbeat control is *only* available in discrete controller implementations.

Perhaps the most noteworthy aspect of these new controllers is that, in their design and operation, a simple time-invariant model of the human operator has been assumed and utilized. Justification for this rather bold move was drawn from fact that the dynamics in question are outside the range of voluntary movement for humans. Further backup is provided by system simulations which model the human input as a constant bias force that exhibit the same chattery behavior which is seen with the old (un-improved) virtual wall controllers. Also inspiring confidence is the fact that these new controllers work so well. Virtual walls implemented with the new controller do not support sustained oscillations where those with the standard controller under the same wall stiffness and sample rate parameters will. In fact, the sampling rate may be substantially reduced or the stiffness substantially increased before the new controllers break down to exhibit irregular or chattery behavior.

The half-sample prediction controller also has an interpretation as a simulation method. It is a constant step-size numerical simulation scheme designed to simulate a discontinuous (two mode) dynamical system without sensitivity to the step size or relative placement of the steps to switching times. Rather than making use of backstepping to locate the switching times between sampling points, it relies on a model of the system (virtual object, interface, and human) in each of its constraint configurations to deduce the switching times from state information available at the sampling points and then account for the effects of a change in configuration occurring at an arbitrary time between the sampling points with an application of deadbeat control.

In summary, although implemented in the sampled data setting, the end effects of the new controller are those of a continuous switching controller–the consequences of the zero-order hold on the control input are not apparent and the effects of intersample switching are quelled.

5.8 Extensions and future work

Natural extensions to the method include the following:

- Lookup tables can be substituted for the functions mentioned above to facilitate speed (ease computational overhead).
- Non-autonomous systems can be handled with minimal error if the external independent agent can be assumed to have bandlimited behavior that is, be predicted 1/2 to 1 sample ahead. To reasonable accuracy with say, a polynomial fit.
- Interaction force could be sensed and used in the prediction
- Improved models of the human might yield better controllers.
- More complex kinematic constraint changes (existing in dynamical models rather than the simple static wall) could be handled by simulation rather than model solution evaluation as used above.
- Perhaps cover constraint changes between human and manipulandum (loss of contact) using similar methods.
- sensor resolution
- use deadbeat control techniques to drive system to known appropriate state when known is derived by other methods, perhaps by examnimation of an integral of the motion, energy conservation, or power exchanges with user, as sensed or calculated.
- computational delays can also be compensated out.

The use of data from a human-characterizing experiment to personalize the rendering of a virtual environment has precedent in the technology of audio environments. A person's head related transfer function (HRTF), which describes the sound filtering properties of their pinnae (outer ears), head, and shoulders as a function of sound source location, can be obtained by comparing a recording made with a small microphone placed in the ear canal with a known sound source in a known location with respect to the person's head. By then filtering a synthesized sound according to that HRTF, that sound can be effectively placed in a position in space (with repsect ot the person's head). Thus, with a particular person's HRTF in hand, an important cue used by humans for localization can be synthesized for that person. These spectral cues are especially imporant for localizing sounds which emanate in the median plane (the plane normal to a line betwen the ears) where inter-aural intensity differences and inter-aural time of arrival differences (the other two cues for localization) play no role.

Depending on the (sometimes) minute topological differences between the pinnae of two persons, a virtual sound environment synthesized with the HRTF of one person may or may not provide successful localization cues for another person. Also, the fact that the HRTF may only be used to place virtual sound sources with repsect to the person's head necessitates real-time head-tracking and real-time filtering. The Convolvotron from Crystal River Corp. are among the best known commercially available products in this field. An associated product is available for the acquisition of HRTFs. See also the work of [] and [] for treatements of this subject.

We have introduced the incorporation of mechanical impedance properties of a person's finger or limb in controller designs for the particularization of the rendering of a virtual haptic environment. Guaranteed performance and optimum perceptual fidelity of synthesized mechanical properties are now possible. We look forward to on-line identification of the human mechanical impedance so that, when a user changes limb posture, properties of the haptic interface would continue to provide maximum impedance-range yet guaranteed passive behavior.

Chapter 6

Analysis of Contact Instability in the Virtual Wall

6.1 Introduction

With the new virtual wall controllers of Chapter 5 in hand, we are ready to make a purchase. Our controllers may be used to buy improved performance for virtual walls —at the expense of a slightly increased computational burden [1]. But before we can judge the prudence of our purchase, we would like to know what our virtual walls stand to gain. Just how much performance improvement may a virtual wall designer expect to enjoy after having paid the price of controller design according to Chapter 5? How bad were the destabilizing effects of the ZOH and the intersample threshold crossing (ITC) in the first place? Our goal for the present chapter, motivated by these questions, is to find measures for the performance improvements afforded by the virtual wall controllers developed in the previous chapter.

As discussed in the introduction to Chapter 5, the problem appears to be a serious one if we are

[1] The increase in computational burden which the controllers of chapter 5 represent (the price to be paid) deserves careful evaluation—it is actually quite small. The zero order hold (ZOH) effect-compensating controller of section 5.4.2 (pole-placement design) requires no extra run-time computation over the standard virtual wall controller —its structure is identical to the standard damped wall controller, only the gain settings are different. To compensate for the ZOH using the half-sample prediction algorithm of section 5.4.1, a certain amount of computational overhead is added at each sampling step. Most of the computations associated with the dead-beat control technique of sections 5.4.3, used to compensate for inter-sample threshold crossing (ITC) may be spread out among the steps within the wall, so that their computational impact is also small.

These new controllers do of course require extra off-line analysis and the execution of a system identification experiment to produce the human impedance model, but these costs do not need to be factored into measures of run-time computational complexity. To locate computations off-line is quite desirable from an engineering viewpoint —delegation is always appropriate when resources are tight.

to judge by the number of virtual wall explorers who have noticed, and researchers who have reported the phenomenon of contact instability. Most every virtual wall designer encounters contact instability as he or she turns up the gain, attempting to create a stiff wall—or turns down the sample rate, attempting to create a computationally efficient wall [2]. Undeniably, contact instability associated with virtual walls remains a serious impediment to further development of virtual environments. It limits the palette of objects which may be placed inside a believable virtual environment.

But we would prefer to judge the magnitude of the problem of contact instability by a quantitative measure other than popular vote. For a gauge of the contact stability problem associated with a particular virtual wall controller, I propose the use of the smallest inherent viscous damping which must be present in the human and/or manipulandum to just guarantee system stability (where the *system* is made up of the human, manipulandum, and virtual wall controller). A finite amount of damping will be required to guarantee system stability of 'standard' virtual walls to compensate for the destabilizing effects of the ZOH and intersample threshold crossing. By contrast, no damping will be required to guarantee system stability for the virtual walls implemented with the controllers of chapter 5, since these controllers, by design, do not suffer the destabilizing effects of the ZOH and intersample threshold crossing. Thus the stabilizing damping coefficients to be associated with the old controllers may be regarded as measures for the performance improvements offered by the new controllers.

Our present task, then, is to size stabilizing damping coefficients. Since the destabilizing effects under scrutiny may be viewed as pathways for the flow of energy into the system, we may equivalently interpret our task as the regulation of the rate of energy dissipation. Note that with the choice of viscous damping for dissipation, we restrict the dissipation rate to be proportional to velocity squared, which may not be the case for the rate of energy introduction. Thus we will have to be cognizant about the degree to which our measure is conservative. Linear damping may not always be the least conservative measure of non-linear energy-introduction.

One caveat remains before launching into a search for damping coefficients. If the zero-order hold effects have already been taken care of, the intersample threshold crossing may either introduce energy or *extract* energy. At wall entry it introduces, at wall exit it extracts. We may expect the two effects to balance—but we cannot prove this supposition without a full analysis. [3] For

[2] Naturally, virtual wall algorithms which can tolerate long sampling periods are highly valued because they allow the comparatively modest computing resources of the computers with which virtual haptic environments are typically implemented (personal computers) to be spent on other time-critical tasks such as graphic updates or networking. Note that personal computers support the hardware interfacing needs of haptic display whereas more powerful computers usually do not

[3] The proof that the energy introduced is balanced by energy extracted will not be undertaken in this chapter. We will stop short of this goal because our problem turns out to be extremely complex. Similar but more tractable problems which have been treated in the field of nonlinear dynamics will be discussed to tentatively infer that a bound on the net energy introduced does in fact *not* exist. In this chapter, we will instead concentrate on *worst-case*

the purposes of finding a measure for the destabilizing effects of intersample threshold crossing, we choose the damping coefficient which will balance the *worst-case* energy introduction. We seek the energy dissipation rate which will balance energy introduced by a full sampling period's delay in turning on and no delay in turning off the wall controller.

For the present, we shall assume that the virtual wall controller is designed to render a wall without virtual damping and that the only means of energy dissipation (aside from the discontinuity in constraint) is through an inherent damper. In this manner we may consider the inherent damping coefficient to be sized for the manipulandum a measure of the destabilizing effects of the discretely implemented virtual wall controller. Extensions to our methods for finding the stabilizing damping coefficient when the virtual wall contains damping will be discussed in the conclusion to this chapter. We will further chose to deal with models of all participants (human, manipulandum, and controller) which are linear except for the discontinuity of the switching wall controller. The behavior we look for when the damper is properly sized (when the energy introduction is perfectly balanced by dissipation is sustained oscillations, indicative of marginal stability. Note that sensitivity to initial conditions wll have to be checked since the energy dissipation rate will certainly be path-dependent, and the energy introduction may be path-dependent in a different manner.

6.1.1 Passivity versus Stability

The answer to the above damper sizing task can only be given with reference to a particular system or at best, class of systems. Yet one participant in the system will always defy modeling and characterization: the human. At issue is just how we will model the human for purposes of analysis, or how we will restrict the set of behaviors which the human may exhibit within our assumed system. We cannot leave the human out of the analysis, since uncoupled stability is not what we are interested in. Indeed, the human finger plays a significant role in determining the stability of this coupled system. Given that the human is capable of many roles, including active behavior, and that as 'audience' or virtual wall explorer, the human should ideally be left free of restrictions or constraining models, it must be acknowledged that our analysis task is quite difficult.

However, as discussed in the previous chapter, virtual wall contact instability is observed without volitional control on the part of the human, that is, when the human can be modeled as an impedance and a bias force. In the interest of minimally restricting the human, in fact to essentially treat the virtual wall and manipulandum without explicit reference to the human impedance, yet guarantee *system* stability, Colgate has called upon the passivity theorem in his recent work [24] and [21]. By assuming that the human remains passive, stability of the *coupled* system may be guaranteed

assumptions.

simply by restricting the ***uncoupled*** controlled manipulandum to remain strictly passive. For the passivity theorem states that the coupling of a passive and a strictly passive system creates a system which will remain stable. An inherent damper can be sized for the manipulandum which will restrict its controlled behavior to that of a strictly passive system, and this damper may be sized without reference to the human impedance. That is, the expression for the lower bound on the inherent damping will not contain any assumed mechanical properties for the human. Auspiciously, this certain inherent damping will guarantee absence of contact instability no matter what human comes up to explore the virtual wall, so long as that human takes on a linear time-invariant (LTI) passive impedance. Colgate's results based on the passivity theorem and their implications for design have proven quite valuable and will be further discussed in the literature review below.

However, Colgate's lower bound on inherent damping to guarantee passivity, which only restricts the human to the class of all LTI passive operators, is not quite pertinent as a measure for the performance improvement offered by our controllers. Our controller designs make use of much more restrictive assumptions about the role of the human in the system. We model the human as a ***particular*** second order impedance, and make use of that model within the controller for prediction. Having adopted the viewpoint that the human can be modeled by a particular impedance, and even folding in the idea of an on-line system identification experiment (to characterize that impedance) into the controller design, Colgate's results based on less restrictive assumptions may be considered conservative for our purposes. We are therefore interested in lower bounds for the inherent damping which stabilize a system which includes a ***particular*** human impedance. In seeking these bounds, we are (once again) considering a problem of stability rather than passivity, and in our presentation of lower bounds, we will be required to make reference to the assumed human impedance.

6.1.2 Outline

In this chapter, we will treat the destabilizing effects of the ZOH and intersample threshold crossing independently. Although we cannot rely on the superposition of the two effects given that we are dealing with a nonlinear system, we are nevertheless interested in separate measures. We would like a measure of the destabilizing effects of the ZOH alone, since we may choose to implement a new controller with ZOH-compensation, but without ITC compensation. Adding the damping coefficient given as a measure of the ITC destabilizing effects to a ZOH-alone compensating controller would account for the ITC effects and thereby guarantee stability. We also undertake the two issues separately in the interest of simplicity of analysis.

The remainder of this chapter is divided into three main sections. Section 6.1 reviews the literature with regard to stability measures for nonlinear systems such as ours. Section 6.2 analyzes the destabilizing effects of the ZOH on certain models. Section 6.3 treats the destabilizing effect of the intersample threshold crossing. Finally, section 6.4 wraps up.

6.2 Literature Review

The use of a smallest damping coefficient as a measure of the destabilizing effects in a system is by no means new. To alter a system with the addition of damping (inherent or virtual) is a natural and reliable means of stabilizing a linear system —it is a standard tool of the controls engineer when a destabilizing effect cannot otherwise be removed. Our choice of viscous damping as a measure of *nonlinear* destabilizing effects is perhaps unique, but probably not to be celebrated, as it is may not be reliable. No claims as to originality or extensibility are being made about the stabilizing damping coefficients found in this chapter. Their intended use, as discussed in the introduction, is for measuring the size of a problem—a problem for which we already have a solution. Basically, these damping coefficients are derived to lend support to statements of usefulness about our solution to the contact instability problem —the new controllers of Chapter 5.

Interestingly, though, Chapter 5 (and by association the present chapter) represent in some significant ways a departure from recent work on the virtual wall. Our contributions to the contact instability problem are new controller designs, whereas most recent work has been centered on analytical treatments of the standard virtual wall controller yielding design guidelines pertaining only to the standard controller. Rather than building on recent analytical results, we have chosen to reject the standard controller and start from scratch on the design problem. Since we now have controllers in hand which are not subject to the energy-introducing effects of the sampled data implementation, we are no longer interested in design guidelines regarding sampled data energetics. Our new designs instead inspire us to undertake some modified analytical treatments of the standard virtual wall. In particular, because our controllers make certain assumptions which were not a part of recent analytical work, we are compelled to revisit this analytical work, incorporate our assumptions, and thereby adapt it for our purposes.

In preparation for the analytical treatments of this chapter, I will give a rather complete review of Colgate's passivity analysis. The subtleties between the results presented here and those presented by Colgate have to do with assumptions of the role of the human in the system (and controller design). The implications of these assumptions lie in degree of conservativeness of results.

A further purpose for covering the literature in such detail is to highlight the one attribute of the standard virtual wall controller which has not been treated explicitly in any published work: the effect of intersample threshold crossing.

6.2.1 Passivity Analysis

Our controllers are designed to work with a particular human impedance, and, when implemented in full, will render stable walls without requiring any extra stabilizing damping from the manipulandum or human. This being the case, the lower bound on inherent manipulandum damping to guarantee passivity determined analytically by Colgate is not quite appropriate as a measure for the performance improvement offered by our new controllers. Rather than incorporating a particular assumed human impedance, Colgate's analyses assumed that the human impedance simply belongs to a class of impedances —the class of all LTI passive impedances. We are now looking for instability measures consistent with the spirit of our new controller designs, that is, which make particular assumptions about the human impedance. We shall be deriving such measures (pertaining to the ZOH effects) with linear stability analyses in the discrete domain.

A comment regarding Colgate's passivity treatment is in order, however. There lies a power in the passivity formalism which makes the results particularly amenable for use as design guidelines. That is, that the damping coefficient which guarantees passivity may be expressed in terms of the transfer function of the controller alone; the human impedance properties are not part of the expression. Our analyses in this chapter, precisely because of our desire to incorporate a particular human impedance, will not take advantage of this power.

Colgate's derivation of the passivity condition for sampled data systems is presented in [24]. Colgate's result is expressed as a lower bound on inherent damping, is based (as is the proof of the passivity theorem) on the small gain theorem. Because the small gain theorem takes only magnitude information into account and completely disregards phase, linear fractional transformations (which have equivalent interpretations as loop transformations and coordinate changes) must be used to reduce conservativeness in applications of the small gain theorem. 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 manipulandum 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 a linear fractional transformation (LFT). A corresponding LFT (coordinate transform) is found for the discrete controller in [21]. Placing the unit disk as a constraint upon this area in the Nyquist plane to which the controller is mapped by this LFT then guarantees coupled stability by the small gain theorem.

A statement of sufficiency is derived via an observation regarding stored kinetic energy. By requiring that the kinetic energy of the mass of the manipulandum never be as great as the total energy input by the human source, the same lower bound on inherent damping is derived as in the necessary condition except that there appears a modulus around the transfer function of the controller.

Colgate's analysis as outlined thus far does not account for the unilateral nonlinearity of the virtual wall. In fact, all components have been assumed linear up to this point. Colgate extends the sufficiency statement to cover a switching controller by making the observation that, since without its feedback controller the manipulandum is passive, if the control input (from the actuator) is set to zero (turned off) at any time, passivity properties will not be affected. This statement does not, however, account for the possible introduction of a small amount of potential energy when the controller is turned *on*. Due to the possibility of crossing the threshold between sampling times, up to a full sampling period T may have elapsed before the wall control law is enacted and the wall will, upon being turned on, effectively hold potential energy without having had the requisite work done on it.

In this chapter we will treat the effects of intersample threshold crossing explicitly.

In an altogether different approach to the contact instability problem, Tsai and Colgate treat the unilateral nonlinearity explicitly in [95]. This analysis, in contrast to that of [24] and [21] is made entirely in the discrete domain. A zero-order-hold equivalent of the plant in feedback connection with a assumed human impedance is found, and used together with a result in filter theory having to do with the saturation non-linearity by Mitra [75].

Rather than via a circle criterion, which uses sector bounds, more complete information about the unilateral nonlinearity is exploited. Tsai and Colgate's results are presented as a Nyquist domain criterion which is reminiscent of the circle criterion. Rather than circles, the forbidden zone for the Nyquist plot of the controller becomes a wedge whose size is frequency-dependent.

In our treatment of the energetics of the intersample threshold crossing, we shall be making use of Poincaré maps—a standard tool in the field of nonlinear dynamics. Although we shall not be contributing to this field, nor making direct use of any theorems from it, I will briefly review similar applications of Poincaré maps and associated stability theorems, especially in the field of robotics.

In the field of robotics, Poincaré maps have been utilized by Koditscheck and Bühler in [17] and [16] to investigate the existence and stability of limit cycles in Raibert's hopping robots and in juggling robots in their own lab. Using some reasonable assumptions regarding the map relating the strike time and strike velocity from one hop to the next can be shown to be one-dimensional. The fact that the map is one dimensional is in fact quite fortuitous— many graphical techniques and related theorems may then be applied. Unfortunately the maps we shall encounter in this chapter having to do with sustained oscillations fed by intersample threshold crossing are two-dimensional and extremely complicated. We shall be making several simplifying assumptions.

Holmes and x have treated the dynamics of a bouncing ball in [47]. Also documented in the popular book by [37]. Their system involved a ball bouncing with a coefficient of restitution on a table which vibrates vertically with a sinusoidal motion. Whether the ball strikes the table in its upward or downward motion is a function of the time spent in the air (an undamped gravity field) which in turn is a function of the last strike time and velocity. As it turns out, this two-dimensional map also reduces to a one-dimensional map. Its dynamics are quite complex, however. Holmes shows the existence of a cascade of period-doubling bifurcations of the limit cycle leading to chaos as the parameters are varied.

Budd and Dux treat a similar system in [15]. But rather than a ball bouncing on a vibrating table, they treat a sprung mass striking a stationary wall where the spring anchor is driven sinusoidally. Theirs is primarily a numerical study.

6.3 Effects of the Sample and Hold

In this section I will develop a measure for the destabilizing influence of the sample and hold operator in our system. The next section will treat the effects of intersample-threshold crossing.

We are interested in sizing an inherent damper within the manipulandum which will guarantee stability when a particular impedance (the human) is coupled, and the virtual wall is simply a spring. We will answer this question with a treatment in the discrete domain. Specifically, a zero-order hold equivalent will be found for the model of the manipulandum together with the human and a linear discrete stability analysis will reveal the damping coefficients.

6.3.1 Uncoupled Stability

Before beginning with the coupled stability analysis, we look at uncoupled stability, in part to outline the procedure.

Our model for the manipulandum is:

$$G(s) = \frac{1}{s^2(ms+b)} = \frac{1}{b}\frac{a}{s^2(s+a)} \tag{6.1}$$

where

$$a = b/m \tag{6.2}$$

The zero-hold equivalent is readily found using a table of Z-transforms, for example, from [31]:

$$G_{zoh}(z) = \frac{1}{b}\frac{z-1}{z}\mathcal{Z}\left\{\frac{G(s)}{s}\right\} = \frac{1}{b}\frac{(aT-1+e^{-aT})z+(1-e^{-aT}-aTe^{-aT})}{a(z-1)(z-e^{-aT})} \tag{6.3}$$

Parameter Values for Figure 6.2

Parameter		Value	Units
m	=	0.30	kg
k	=	500	N/m

In feedback connection with a controller $H(z) = K$, as in Figure 6.1, our closed loop characteristic equation becomes:

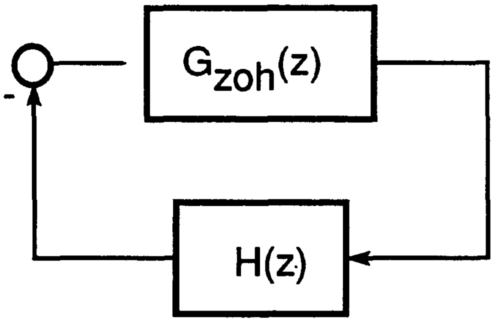


Figure 6.1: *Feedback connection between manipulandum and controller, no human impedance coupled*

$$z^2 + \left[\frac{K}{ab}aT - (\frac{K}{ab} + 1) + (\frac{K}{ab} - 1)e^{-aT}\right] z + \left[\frac{K}{ab} - (\frac{K}{ab} - 1)e^{-aT} - \frac{K}{ab}aTe^{-aT}\right] = 0 \qquad (6.4)$$

We seek the values of b which place the roots of the characteristic equation on the unit circle. These may be found by setting the last term of the previous equation equal to unity.

This produces the equation:

$$e^{-bT/m}\left[1 - \frac{KT/b}{(1 - mK/b^2)}\right] = 1 \qquad (6.5)$$

The damping coefficients which produce marginal stability may be found numerically and plotted as a function of the sampling period, as in Figure 6.2. Values have been assumed for each of the parameters in producing this plot, as shown in Table 6.3.1

The results for uncoupled stability may be presented in non-dimensional parameters as suggested

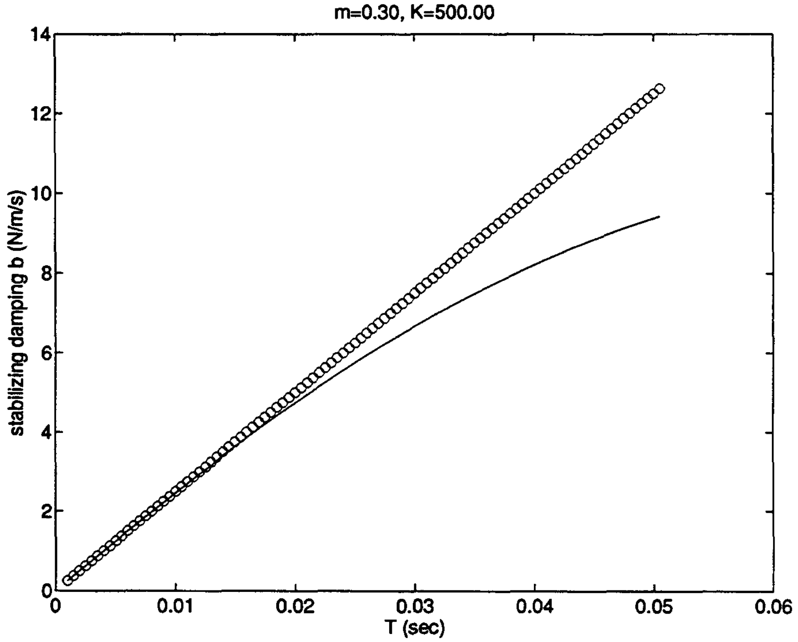


Figure 6.2: *Stabilizing Damping versus Sampling Period*

by Colgate (See [24]):

$$\tau = bT/m \quad \alpha = kT/b \tag{6.6}$$

Equation 6.5 reads, in non-dimensionalized parameters:

$$e^{1/\tau} = 1 - \frac{\alpha}{1 - \tau\alpha} \tag{6.7}$$

We may plot this alongside the passivity region derived by Colgate in a graph of α versus τ, which has been done in Figure 6.3. Note that the parameter space for uncoupled stability is larger than that for passivity.

A simulation may be used to check marginal stability of a borderline damping coefficient. Selection of the point circled in Figures 6.2 and 6.3, for example, produces marginal stability as expected: See Figure 6.4.

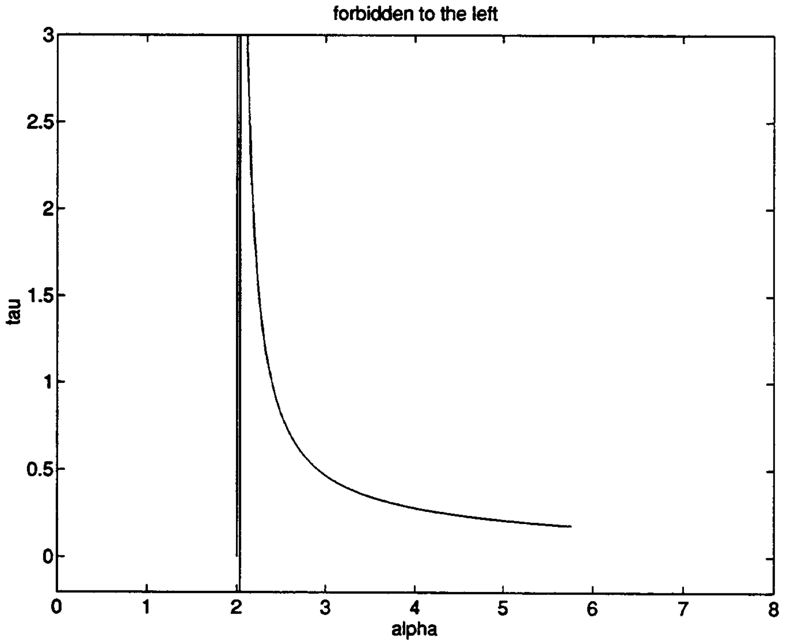


Figure 6.3: *Alpha versus Tau*

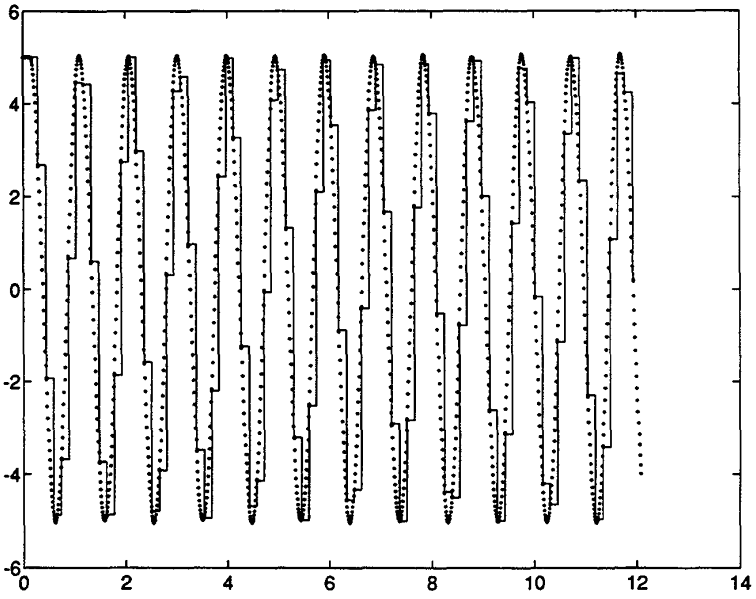


Figure 6.4: *Verification of marginal stability by simulation*

6.3.2 Coupled Stability

To treat stability when the manipulandum is coupled to a particular impedance, we may follow the same basic procedure after making a reasonable assumption about the manner in which the human impedance is coupled to the manipulandum. We shall assume that the mechanical coupling between the effective mass of a human finger and the mass of the manipulandum is direct, as shown in Figure 6.5. We neglect the compliance of the fingertip skin in making this assumption. These assumptions are consistent, however, with experimental measures of the finger impedance as in the work of Hajian and Howe [39]. Thus the human impedance does not increase the order of our system, and therefore analysis is almost as simple as in the uncoupled case. A block-diagram interpretation of the assumed human impedance/ manipulandum coupling is shown in Figure 6.6

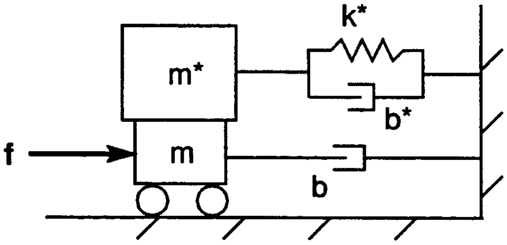


Figure 6.5: *Assumed mechanical coupling between modeled human impedance and manipulandum*

We assume a second order linear impedance to model the human:

$$Z_o(s) = m^* s^2 + b^* s + k^* \tag{6.8}$$

Coupling between the human and manipulandum produces the following expression for the composite impedance:

$$G^*(s) = \frac{1}{(m + m^*)s^2 + (b + b^*)s + k^*} \tag{6.9}$$

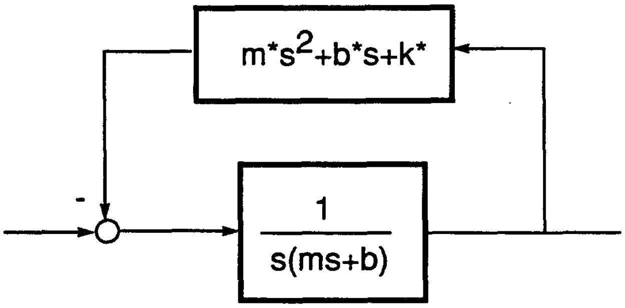


Figure 6.6: *Block diagram interpretation of assumed human-manipulandum mechanical coupling*

The zero order hold equivalent of $G^*(s)$ is:

$$G^*_{zoh}(z) = \frac{1}{k^*}\frac{z-1}{z}\mathcal{Z}\left\{\frac{G^*(s)}{s}\right\} = \frac{1}{k^*}\frac{Az+B}{z^2 - 2e^{-aT}(cosbT)z + e^{-2aT}} \tag{6.10}$$

where

$$\begin{aligned} A &\triangleq 1 - e^{-aT}cosbT - \tfrac{a}{b}e^{-aT}sinbT \\ B &\triangleq e^{-2aT} + \tfrac{a}{b}e^{-aT}sinbT - e^{aT}cosbT \end{aligned} \tag{6.11}$$

A feedback connection between this transfer function and the simple controller $H(z) = K$ produces the closed-loop characteristic equation:

$$z^2 + \left[\frac{K}{k^*}A - 2e^{-aT}(cosbT)\right]z + \left[\frac{K}{k^*}B + e^{-2aT}\right] = 0 \tag{6.12}$$

To set the modulus of the roots equal to unity, we set the last term to unity, or, with substitution of the definition of B,

$$\frac{K}{k^*}\left(e^{-aT} + \frac{a}{b}e^{-aT}sinbT - e^{aT}cosbT\right) + e^{-2aT} = 1 \tag{6.13}$$

Roots may be found analytically and plotted versus various parameters. In Figure 6.7 we show the dependence of b on the sampling period T, along with the dependence of the stabilizing damping

Parameter Values for Figure 6.7

Parameter		Value	Units
m	=	0.30	kg
k	=	5000	N/m
m*	=	0.02	kg
b*	=	3.75	N/m/s
k*	=	257	N/m

coefficient in the uncoupled case, from Figure 6.2. Particular values have been chosen for each of the parameters, including the parameters for the assumed human impedance for this plot, as shown in Table 6.3.2. Note that the positive damping b^* contributed by the human impedance provides for the possibilty of adding negative damping for low sampling periods.

Unfortunately, these data do not lend themselves to presentation in non-dimensionalized variables.

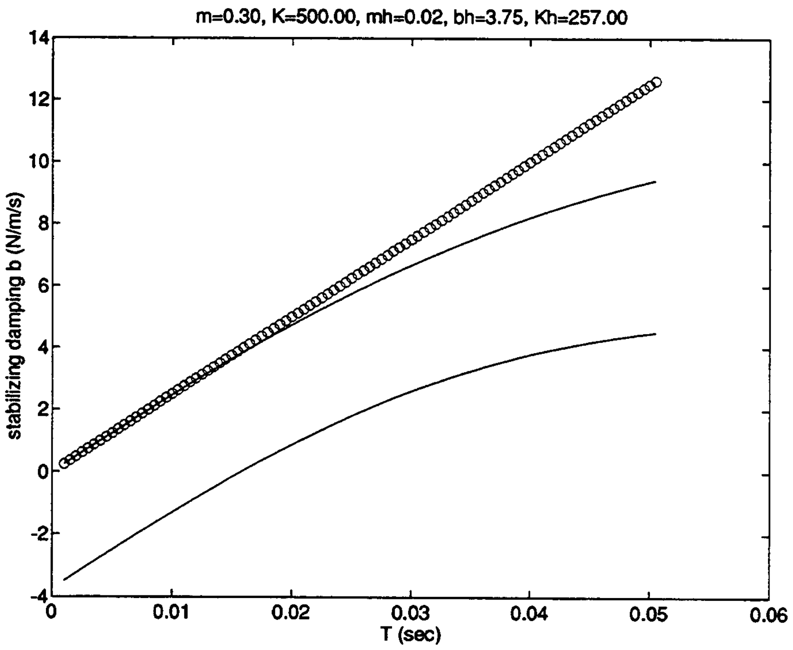


Figure 6.7: *Stabilizing Damping versus Sampling Period*

6.4 Effects of the intersample threshold crossing

In this section, we will evaluate the negative impacts of the intersample threshold crossing upon the stability of a virtual wall-rendering manipulandum coupled to a particular human. From Chapter 5, we have a controller design in hand which, with deadbeat control techniques, is able to eliminate the effects of intersample threshold crossing, even while working within the constraints of a sampled data controller. But before we bother to implement this special control technique, we would like to know what we will be gaining and thus undertake a bit of analysis of a controller which does not use this technique. We will be treating the destabilizing effects of the intersample threshold crossing (ITC) independently of the zero order hold (ZOH) and will thus produce a separate stabilizing damping coefficient. An appropriate use for this ITC-effect balancing damper would be as a safety margin in damping to add to a system which uses ZOH compensation but not ITC compensation.

In order to treat ITC independently of the ZOH, we will set up the following system for analysis: a continuous but switching controller (two modes) in which the switching times must fall on integer multiples of a fixed sampling period T.

$$t_{on},\ t_{off} \in \{jT \,|\, j \in \mathcal{Z}\} \tag{6.14}$$

The first detection of a position beyond the switching line, occurring on a sampling time, will trigger a switch. Thus the latency in switching can last up to one full sample period T. Once the switch has been thrown, the controller operates like an analog controller, without sampling and without a zero-order held controller output.

Standard laws of conservation of energy do not apply to time-varying systems. For example, discontinuously increasing the stiffness of a compressed spring will all of a sudden increase that spring's stored energy. The kinetic energy of a moving mass will abruptly increase at time t if the mass increases at time t. Discontinuously changing the stiffness of an uncompressed spring, or the mass of an unmoving object, however, cannot add energy. This fact has been used to advantage in the design of physical models for sound synthesis by Van Duyne and Pierce [99]. An evolving spectrum can be effected in a physically modeled string by using a nonlinear spring for the model of the bridge. If the spring discontinuously changes spring constants, but does so only when unextended, it will cause the spectral energy to rise in frequency without affecting the damping characteristics. Evolving spectra are characteristic of some musical instruments, for example the gong. Van Duyne and Pierce call such model components 'passive nonlinearities'.

Our system contains an element which would have been considered a passive nonlinearity, had it not been for the latency in switching times. Nominally, (if switching times were *on* threshold crossings), the system could not gain energy since the switching would, in that case, occur when the spring of the virtual wall is uncompressed.

Since we cannot implement our would-be passive nonlinearity in a sampled data setting, we now ask the following questions:

First, if energy is introduced by switching on a spring in a compressed state (as takes place at wall entrance), and energy is extracted by only first turning off the spring in a stretched state (at wall exit), can we say that the energy introduced is balanced by the energy extracted? Or do there exist pathological cases in which the net energy continually increases, resulting in unbounded growth of the state variables?

Second, under a worst case assumption, where maximum energy is introduced for every wall strike, what damping coefficient must be included in our system to guarantee stability? When the manipulandum strikes the wall (the threshold is crossed) just at that time between sampling times which makes for the worst latency, what is the largest discontinuous jump in potential energy in the spring?

To answer these questions, we make use of Poincaré maps. Poincaré maps (also called return maps) are often used in the field of nonlinear dynamics to treat discontinuous (switching) systems. By expressing the dynamics of a switching system in a return map, the switching dynamic effectively disappears. The switching analog system is transformed into a non-switching discrete system. As a time-invariant discrete system model, the return map lends itself to analysis by standard tools from linear discrete systems theory.

To develop a full appreciation for the complexity of our system, especially to point out that the net amount of energy introduced (or extracted) is a function of both the time and the state at each strike, I first present a full return map, without simplifying assumptions. Thereafter, section 6.4.4 will present a simplified map and its associated assumptions. The full return map will only be useful for numerical studies, whereas the one-dimensional map will lend itself to analytical treatment.

6.4.1 Full return map

To formulate a return map, we choose the Poincaré section at $y = 0$ (the wall threshold) and further choose, from the two crossings per bounce, only the crossing of this section in the positive y direction (the wall exit). Thus our return map will be a composite of two maps: F_1, which maps the system state from wall exit to wall entry, and F_2 which maps the system state from wall entry to the next wall exit.

$$F = F_1 \circ F_2 \tag{6.15}$$

Figure 6.8 defines the time points and time intervals which will enter into our discussion during

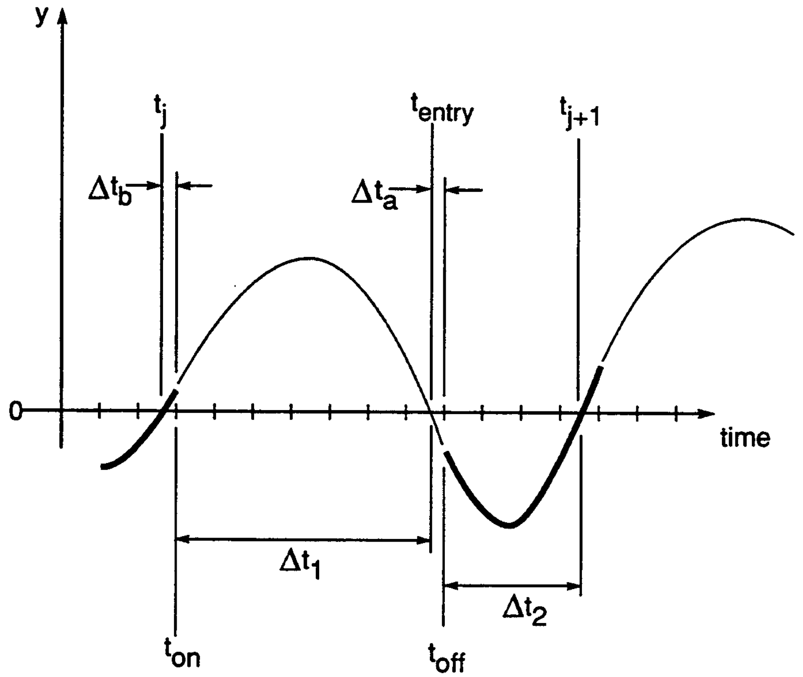


Figure 6.8: *Sampling Points in a Typical floor Strike*

the construction of F_1 and F_2. Note that the map F_1 and F_2 shall both express evolution according to the *wall off* and *wall on* models, because of the asynchrony between threshold crossings and switching times and our choice of the threshold as the Poincaré section. Primarily, however, F_1 expresses evolution under the *wall on* and F_2 expresses evolution under the *wall off* model.

We shall only need to carry two variables to characterize the switching sequence, namely t_j and v_j, since the position y_j is zero at each threshold crossing (the chosen Poincaré section). Thus the map F is made up of two functions f and g:

$$F: \begin{cases} t_{j+1} = f(t_j, v_j) \\ v_{j+1} = g(t_j, v_j) \end{cases} \tag{6.16}$$

To be fully explicit, we show F_1 and F_2 cast in the same form below.

$$F_1 : \begin{cases} t_{entry} = f_1(t_j, v_j) \\ v_{entry} = g_1(t_j, v_j) \end{cases} \tag{6.17}$$

$$F_2 : \begin{cases} t_{j+1} = t_{exit} = f_2(t_{entry}, v_{entry}) \\ v_{j+1} = v_{exit} = g_2(t_{entry}, v_{entry}) \end{cases} \tag{6.18}$$

Two models will govern the motion of the ball, the ball being our model of the manipulandum under the finger or hand of the human. One will govern during the *wall on* periods of motion, to be known as Model I, and another will govern during the *wall off* periods, known as Model II. Various model types may be used for Model I (including a ball falling freely or with damping in a gravity field or a lightly-sprung mass with or without damping and gravity), so long as the ball, moving according to such model, will return to the wall in finite time. Note that the gravity field represents a constant bias force from the hand of the human operator. Model II will generally take the form of a sprung mass, with or without damping and gravity, wherein the spring represents the virtual wall stiffness. The following development will assume that explicit solutions exist to both Model I and Model II. But before assuming particular models, I will construct the full return map F without reference to particular models. Models will be assumed just before introducing the numerical studies.

We will require the following four functions, constructed from the solutions of Model I and Model II.

First, function $y_i : \Re^2 \times \Re \rightarrow \Re^2$ returns the state x, (where $x = [y, v]'$) which results from evolution according to Model I from the initial condition x_0:

$$\mathbf{x} = y_i(\mathbf{x_0}, t) \tag{6.19}$$

Second, we will require a function $t_i : \Re^2 \rightarrow \Re$ which returns the time remaining to wall strike employing Model I given an initial condition:

$$\Delta t_1 = t_i(\mathbf{x_0}) \tag{6.20}$$

Third, function $y_{ii} : \Re^2 \times \Re \rightarrow \Re^2$ is used to evolve the state according to Model II:

$$\mathbf{x} = y_{ii}(\mathbf{x_0}, t) \tag{6.21}$$

Finally, a function $t_{ii} : \Re^2 \rightarrow \Re$ will return the time remaining to wall strike employing Model II given an initial condition:

$$\Delta t_2 = t_{ii}(\mathbf{x_0}) \tag{6.22}$$

We begin with the construction of F_1.

Although the ball has just passed the threshold, exiting the wall at t_j, it continues to travel according to Model II (*wall on*) to the end of the sample period containing t_j. The time t_{off} at which Model I takes over from Model II is given by:

$$t_{off} = T(1 + floor\left[\frac{t_j}{T}\right]) \tag{6.23}$$

where the *floor[]* function returns the largest integer less than its argument.

We use Model II to evolve the state from $x_j = [0, v_j]'$ at t_j to x_{off} at t_{off}.

$$x_{off} = y_{ii}(x_j, t_{off} - t_j) \tag{6.24}$$

The time to wall strike is found from the state x_{off} using function t_i:

$$\Delta t_1 = t_i(\mathbf{x_j}) \tag{6.25}$$

We already have the two components of F_1, f_1 and g_1:

$$\begin{cases} t_{entry} = t_{off} + \Delta t_1 \\ x_{entry} = y_i(\mathbf{x}_{off}, \Delta t_1) \end{cases} \tag{6.26}$$

The position component of x_{entry} will be zero, only v_{entry} is required for the map.

Now for the construction of F_2.

The time at which Model II takes over from Model I, t_{on} occurs on the first sampling time after t_{entry}, and is found with:

$$t_{on} = T(1 + floor\left[\frac{t_{entry}}{T}\right]) \tag{6.27}$$

The state at t_{on} is available using function y_i:

$$x_{on} = y_i(x_{entry}, t_{on} - t_{entry}) \tag{6.28}$$

To bring the state to time $t_{j+1} = t_{exit}$, we require the time spent in Model II:

$$\Delta t_2 = t_{ii}(x_{on}) \tag{6.29}$$

We now may express the time and state at t_{j+1}:

$$\begin{cases} t_{j+1} = t_{on} + \Delta t_2 \\ x_{j+1} = y_{ii}(x_{on}, \Delta t_2) \end{cases} \tag{6.30}$$

The position y_{j+1} will be zero. The threshold crossing sequence requires only v_j and t_j.

6.4.2 Substitution of models

Unfortunately, the full return map does not yield itself to further analysis since it includes the non-analytic floor function. The floor function serves to place the switching times properly on the sampling times, always choosing the next sampling time after a threshold crossing. But the presence of the floor function is not the only property which makes this map difficult to analyze. The other property is the coupling between the two component functions f and g. The latency in switching between models has an effect on the ensuing threshold crossing velocity. Thus the exit velocity v_{j+1} is a function (denoted g) of both t_j and v_j. Likewise, the time period for which the ball remains in the wall is a function of the velocity at wall strike, and thus t_{j+1} is a function (denoted f) of v_j as well as t_j. Thus we have a two-dimensional map, for which far fewer tools are available than for one-dimensional maps.

The utility of this full return map lies in efficient numerical studies. To demonstrate one such numerical study, and thereby further highlight the interesting complexity of our system, I will assume two simple undamped models for Model I and Model II and use them to run numerical simulations.

Model I will take the form of a ball falling freely in a gravity field. Function y_i (Equation 6.19) thus reads:

$$\begin{aligned} y &= v_0 t - \tfrac{1}{2} g t^2 \\ v &= v_0 - gt \end{aligned} \tag{6.31}$$

Function t_i (Equation 6.20) using the freely falling ball model reads:

$$\Delta t_1 = \frac{-v_0 + \sqrt{v_0^2 + 2gy_0}}{g} \tag{6.32}$$

Function y_{ii} (Equation 6.21) using a model of a sprung mass without damping reads:

$$\begin{aligned} y &= Acos\omega t + Bsin\omega t - g/\omega^2 \\ v &= -(\omega y_0 + g/\omega) sin\omega t | v_0 cos\omega t \end{aligned} \tag{6.33}$$

where $\omega = \sqrt{k/m}$, $A = (y_0 + g/\omega^2)$ and $B = v_0/\omega$.

Finally, the function t_{ii} (Equation 6.22) reads:

$$\Delta t_2 = \frac{1}{\omega}\left[\pi + atan2(A, -B) + asin\left(\frac{g/\omega^2}{\sqrt{A^2+B^2}}\right)\right] \tag{6.34}$$

The return map defined above, used with these function definitions, yields a very efficient means of computing the sequence of threshold crossings.

The map is simply too complicated to make any analytical deductions regarding the balance of energy gain with energy loss. However, we may use it to conduct numerical studies into these questions. Intuitively, we may expect that as the samping period decreases, the amount of energy gained because of intersample threshold crossing will decrease. We have tested this hypothesis as follows. The map was used to compute the exit velocity for 40 strikes of the wall, using various sampling periods. The maximum strike velocity of those 40 is ploted versus the sampling period in Figure 6.9 The sampling period was varied in increments of .0001 from 0.01 to 0.3 seconds.

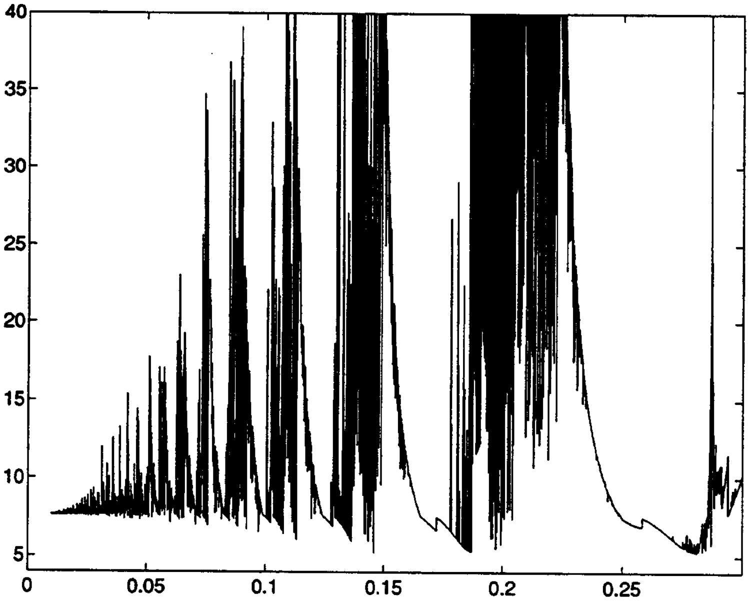


Figure 6.9: *Maximum exit velocity of 40 strikes versus sampling period*

We observe that the maximum velocity attained does seem to decrease with decreasing sampling period. But perhaps more interesting is just how non-predicatable the situation really is.

6.4.3 Simulations and Checks on the Return Map

I will briefly describe the manner in which we have verified the return map against simulations and evaluations of model solutions. This section will provide a transition to the next, in which simplifying assumptions render the return map analytically tractable.

Model simulations may be run according to algorithms which account for the sampled data implementation such as those introduced in Chapter 5. Alternatively, since we have assumed models which posess solutions, it is only necessary to evaulate the solutions according to an appropriate switching algorithm to produce time histories of the system. As a further check, we have found expressions for the kinetic energy, potential energy and dissipated energy as a function of state and history for various models. Highlighting our verification procedure here, and taking this opportunity to present plots of the kinetic, potential, and dissipated energy will further exemplify the coupled and complex nature of this nonlinear system.

Figure 6.10a) shows the Kinetic Energy KE, Potential Energy PE and cumulative dissipated energy E_d in a time chart above the position and velocity trajectories of Figure 6.10b). These plots pertain to the full sampled data simulation containing the *floor* function. Note that the potential energy (stored in the spring and gravity field) jump discontinuously at wall entry and exit. Energy is gained at wall entry and lost at wall exit, to differing degrees. The energy dissipated may never catch up to the total energy. The ball may not stop bouncing, despite the presence of dissipation in this model.

By contrast, Figure 6.11 shows the state evolution and energy evolution in the case where the switching times are *on* the threshold crossings. In this figure, which corresponds to *physical* behavior, or the successful implementation of a passive nonlinearity, we see that the total Energy is conserved.

We now turn to an analysis of worst-case scenarios.

Figure 6.12 shows the energy and state evolution of the worst-case system without bias with the balancing damping b_{bal}.

A phase-plot of the worst-case scenario is also informative as in Figure 6.13. The growth in state due to delayed switching can be clearly seen. Delayed switching across the threshold (switching in the first quadrant at wall exit and in the third for wall entry) causes the state trajectory to jump onto paths which are further and further from the origin.

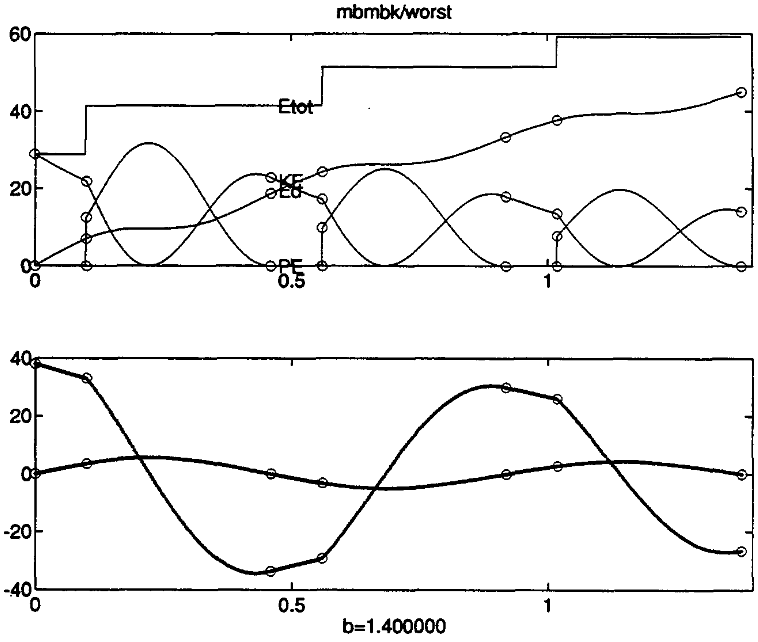


Figure 6.10: *Floored*

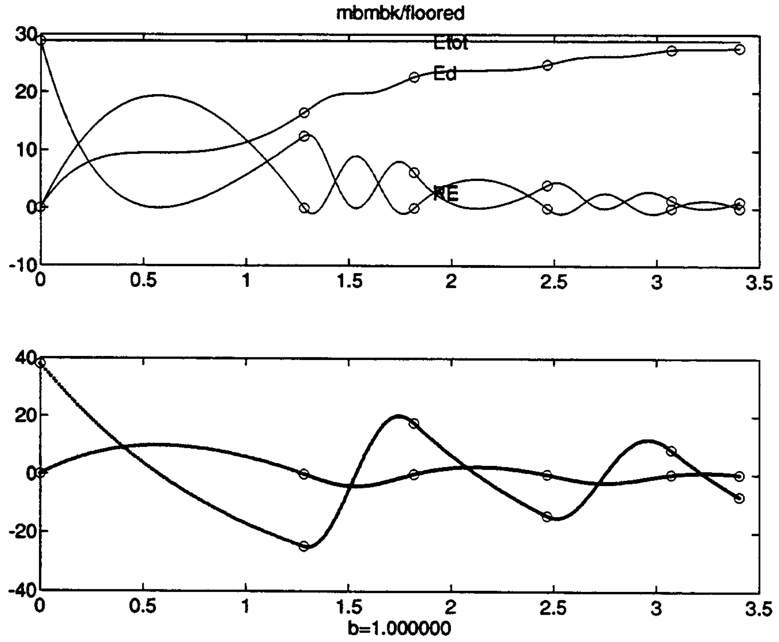


Figure 6.11: *Physical*

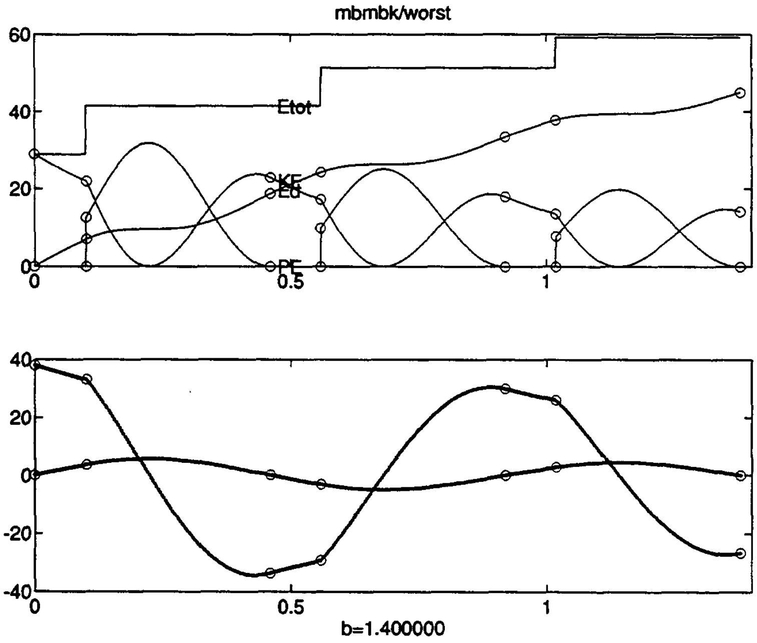


Figure 6.12: *Worst-Case*

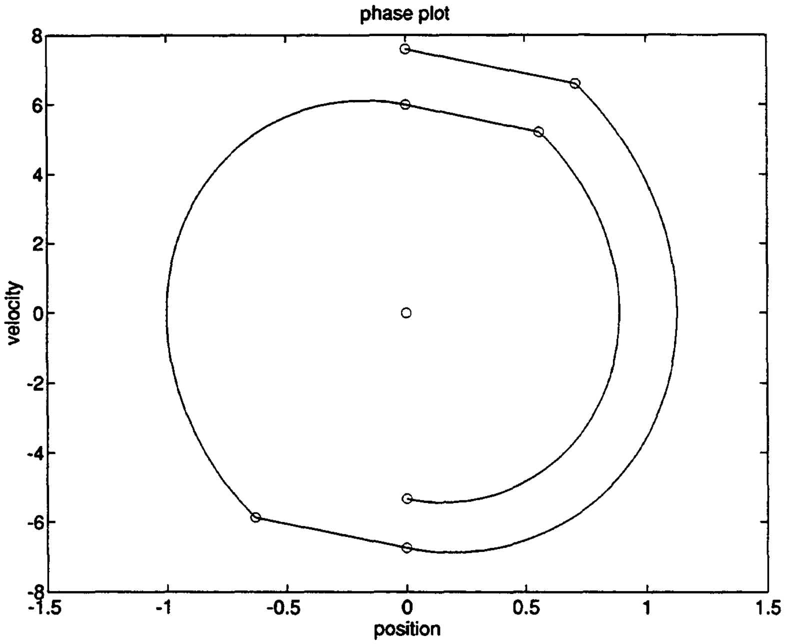


Figure 6.13: *Worst-Case Phase Plot*

6.4.4 Effects of intersample threshold crossing under worst-case assumptions and non-existent bias force

The worst case energy gain obtains when our bouncing ball hits the wall (crosses the threshold) immediately after a sampling time. In this manner, the ball will pierce the wall to a maximum depth while traveling according to Model I —using an entire sampling period. The minimum energy loss obtains when the ball is held back by the spring of the wall upon exit for minimum time. Thus the wall turn-on switching time occurs T seconds after wall entry and the wall turn-off switch occurs right on wall exit in the worst-case scenario.

Under this worst case assumption we may seek the damping coefficient b which will dissipate the energy gained. This is of course a much simpler problem than that of the previous section, where the *floor* function was involved. In the present case, if the strike velocity is the same from one wall encounter to the next (as will be the case when the balancing damping b_{bal} has been found), then the energy gain will be the same from one encounter to the next. This was not the case in the full return map because the energy gained at each bounce had to do with both the strike time and strike velocity.

The balancing damping coefficient, however, still may not be found analytically until we make one further assumption. Unfortunately, no analytical solutions exist to the following deceptively simple equation:

$$e^{a\theta} cos(\theta) = C \tag{6.35}$$

The roots of this equation are the intersection of a logarithmic spiral with an off-axis vertical line. Its solution is necessary to express the time to exit the wall given the wall entry velocity. The constant on the right-hand-side arises from the non-zero bias force of gravity bearing down. In full form, we seek the time Δt_2 which solves the following equation:

$$y(\Delta t_2) = e^{a\Delta t_2}(Acos\omega\Delta t_2 + Bsin\omega\Delta t_2) - g/\omega_0^2 = 0. \tag{6.36}$$

Beyond a certain time, the curve of an exponential spiral will no longer intersect an off-center line. Thus the solution we seek is not a periodic function.

We therefore make one more simplifying assumption. We set the bias force to zero ($g = 0$) and further, to ensure that the ball will return to the wall in finite time, we use a sprung mass for Model I. We set the spring constant on Model I, K_1, much lower than that on Model II, retaining the wall switching characteristic.

We may now treat our problem analytically, developing the map as follows:

Function y_i, which expresses evolution of Model I (Equation 6.19) takes the form:

$$\begin{aligned} y_1(t) &= e^{-\sigma t}(A_1 cos w_1 t + B_1 sin w_1 t) \\ v_1(t) &= e^{-\sigma t}\left[(B_1 w_1 - A_1\sigma)cos\omega_1 t - (B_1\sigma + A_1\omega)sin w_1 t\right] \end{aligned} \tag{6.37}$$

The initial conditions are used to define A and B:

$$\begin{aligned} y_1(0) = y_{10} = 0 &\Rightarrow A_1 = 0 \\ v_1(0) = v_{10} &\Rightarrow B_1 = v_{10}/\omega_1 \end{aligned} \tag{6.38}$$

The time spent in Model I (function t_i, Equation 6.20) is given simply as:

$$\Delta t_1 = \pi/\omega_1 \tag{6.39}$$

The function y_{ii} (Equation 6.21) is similar to function y_{ii}, but with a larger spring stiffness K_2, and thus larger natural frequency ω_2:

$$\begin{aligned} y_2(t) &= e^{-\sigma t}(A_2 cos w_2 t + B_2 sin w_2 t) \\ v_2(t) &= e^{-\sigma t}\left[(B_2 w_1 - A_2\sigma)cos\omega_2 t - (B_2\sigma + A_2\omega)sin w_2 t\right] \end{aligned} \tag{6.40}$$

Constants A_2 and B_2 are evaluated using the initial conditions:

$$\begin{aligned} y_2(0) = y_{20} &\Rightarrow A_2 = y_{20} \\ \dot{y}_2(0) = v_{20} &\Rightarrow B_2 = v_{20}/\omega_2 \end{aligned} \tag{6.41}$$

The time spent in model II is handled by function t_{ii} (Equation 6.22), defined as follows:

$$\Delta t_2 = (1/\omega_2)atan2(-A_2, B_2) \tag{6.42}$$

With the functions y_i, t_i, y_{ii}, and t_{ii} in hand, we may develop an explicit expression for the return map in a manner similar to the development of the full return map above.

The time spent in Model I is simply one sampling period added onto the time spent outside the wall:

$$\Delta t_1 = \pi/\omega_1 + T \tag{6.43}$$

The state at wall entry may then be found using Δt_1 in Equation 6.37

$$\begin{aligned} y_{20} = y_1(\Delta t_1) &= e^{-\sigma \Delta t_1}\tfrac{v_{10}}{\omega_1} sin(\omega_1 \Delta t_1) \\ v_{20} = v_1(\Delta t_1) &= e^{\sigma \Delta t_1}\left[v_{10} cos(\omega_1 \Delta t_1) - \tfrac{v_{10}}{\omega_1}\sigma sin(\omega_1 \Delta t_1)\right] \end{aligned} \tag{6.44}$$

The time spent inside the wall is given by:

$$\Delta t_2 = \frac{1}{\omega_2} tan^{-1}\left(\frac{-A_2}{B_2}\right) = \frac{1}{\omega_2} tan^{-1}\left(\frac{-y_{20}}{(1/\omega_2)(v_{20} + y_{20}\sigma)}\right) \tag{6.45}$$

After substituting for y_{20} and v_{20} from equation 6.44, this equation simplifies to:

$$\Delta t_2 = \frac{1}{\omega_2} tan^{-1}\left(\frac{\omega_2}{\omega_1} tan(\omega_1 t_1)\right) \tag{6.46}$$

We have, since Δt_2 is the time to wall threshold,

$$y_2(\Delta t_2) = e^{-\sigma \Delta t_2}\left[A_2 cos\omega_2 \Delta t_2 + B_2 sin\omega_2 \Delta t_2\right] = 0 \tag{6.47}$$

The velocity at wall exit is expressed using Δt_2 in Equation 6.40:

$$v_2(\Delta t_2) = e^{-\sigma t_2}\left[(B_2 w_1 - A_2\sigma)cos\omega_2 t_2 - (B_2\sigma + A_2\omega)sin w_2 t_2\right] \tag{6.48}$$

re-arranging, we have:

$$v_2(\Delta t_2) = e^{-\sigma t_2}\left[-\sigma(A_2 cos\omega_2 t_2 + B_2 sin\omega_2 t_2) + \omega_2(B_2 cos\omega_2 t_2 - A_2 sin\omega_2 t_2)\right] \tag{6.49}$$

but, from Equation 6.47, the first term is zero. So

$$v_2(\Delta t_2) = e^{-\sigma t_2}\omega_2\left(B_2 cos\omega_2 t_2 - A_2 sin\omega_2 t_2\right) \tag{6.50}$$

$$= e^{-\sigma t_2}\omega_2\left(\frac{1}{\omega_2}e^{-\sigma t_1}v_{10}cos(\omega_1 t_1)cos\omega_2 t_2) - e^{-\sigma t_1}\frac{v_1 0}{\omega_1}sin(\omega_1 t_1 sin(\omega_2 t_2)\right) \tag{6.51}$$

$$= e^{-\sigma(t_1+t_2)}v_{10}\left(cos(\omega_1 t_1)cos(\omega_2 t_2) - \frac{\omega_1}{\omega_2}sin(\omega_1 t_1)sin(\omega_2 t_2)\right) \tag{6.52}$$

The return map may now be expressed as follows:

$$\begin{aligned} t_{j+1} &= t_j + \Delta t_1 + t_1 + t_2 \\ v_{j+1} &= e^{-\sigma(t_1+t_2)}v_j\left[cos\omega_1 t_1 cos\omega_2 t_2 - \tfrac{\omega_2}{\omega_1}sin\omega_1 t_1 sin\omega_2 t_2\right] \end{aligned} \tag{6.53}$$

where

$$\begin{aligned} t_1 &= \tfrac{\pi}{\omega_1} + T \\ t_2 &= \tfrac{1}{\omega_2}tan^{-1}\left(\tfrac{\omega_1}{\omega_2}tan\omega_1 t_1\right) \end{aligned} \tag{6.54}$$

Note that our two functions f and g are decoupled:

$$\begin{cases} t_{j+1} = f(t_j) \\ v_{j+1} = g(v_j) \end{cases} \tag{6.55}$$

We may find the Jacobian of the map

$$J = \begin{bmatrix} \frac{\partial f}{\partial t} & \frac{\partial f}{\partial v} \\ \frac{\partial g}{\partial t} & \frac{\partial g}{\partial v} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & e^{-\sigma(t_1+t_2)} \left(cos(\omega_1 t_1) cos(\omega_2 t_2) - \frac{\omega_1}{\omega_2} sin(\omega_1 t_1) sin(\omega_2 t_2) \right) \end{bmatrix} \tag{6.56}$$

Setting the determinant of the Jacobian equal to 1, we find the limit cycles (fixed points) are given by:

$$\sigma = \frac{1}{t_1 + t_2} ln \left(cos(\omega_1 t_1) cos(\omega_2 t_2) - \frac{\omega_1}{\omega_2} sin(\omega_1 t_1) sin(\omega_2 t_2) \right) \tag{6.57}$$

This equation may be solved numerically for b_{bal}, the balancing damping coefficient.

Figure 6.14 shows the balancing b_{bal} as a function of the period T under the worst case assumpion.

As mentioned above, solutions are not available when $g \neq 0$, but they may still be found numerically. Having a semi-analytic result in hand for the case $g = 0$ allows us to proceed with a bit more confidence.

Figure 6.15 shows a three-dimensional plot with the analytical solution highlighted. The bias force (gravity) increases and is seen to have a increasing influence on the damping coefficient b, but not as strong as the sampling period T. (Within the range shown).

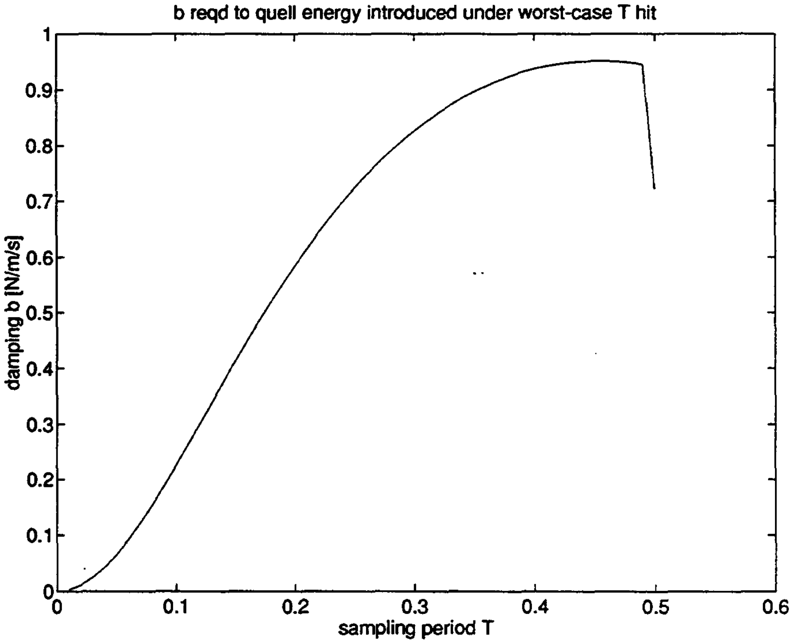


Figure 6.14: *Damping required to quell the energy introduced under worst-case assumptions*

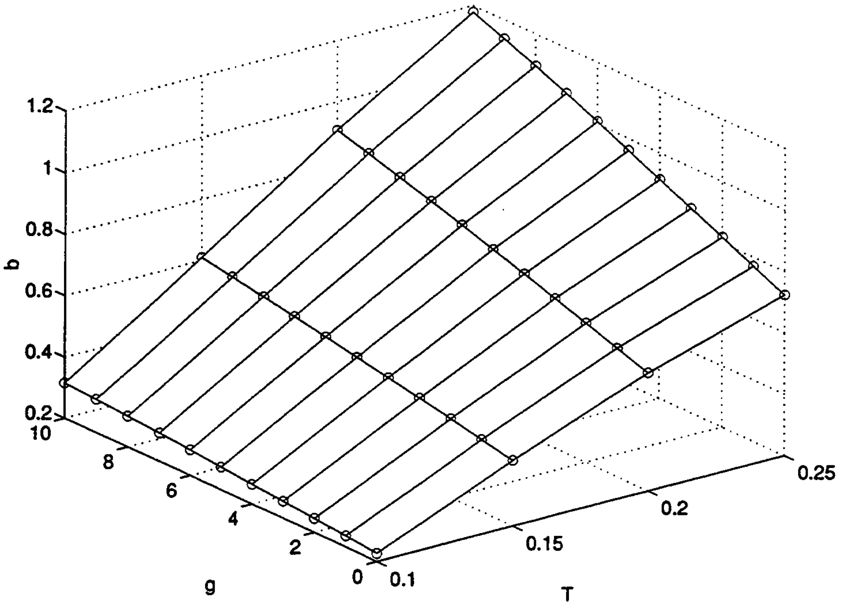


Figure 6.15: *Stabilizing damping as a function of both samping period T and bias force g*

6.5 Summary

In this chapter, measures for the destabilizing effects of the zero order hold and the intersample threshold crossing have been found which are useful in determining the worthiness for implementation of the control techniques introduced in the previous chapter. An analytical solution for the damping coefficient which will balance the destabilizing effects of intersample threshold crossing under worst-case and non-existant bias force from the human was presented. Numerical extensions of this result were made with a Poincaré map.

Chapter 7

Summary and Future Work

The work of this thesis was undertaken with the ambitious plan of creating a motorized synthesizer keyboard which would feel and behave like the keyboard of a grand piano. To a significant extent, we have accomplished this goal. We now have in hand a keyboard of seven keys which exhibits some important aspects of the dynamical response (in terms of both mechanical impedance and sound) of the grand piano. We have arrived at a unique position, ready to design and carry out psychophysical experiments which test the original hypotheses underlying our project —that appropriate force feedback from a synthesizer leads to increased potential for musical expression.

7.1 Looking Back

The path to our present position has included several steps, each of which occupy, in description, a chapter of this thesis.

Psychophysics

First, we placed our goals with regard to the role of force feedback and auditory feedback from a keyboard instrument in clear perspective. Design of a force-reflecting device for musical control entails many subtleties in a field which remains largely unexplored. We are interested in forging ahead in the field of haptic interface despite the fact that human processing of haptic information is not yet fully understood. Satisfactory conclusions have not been reached as to which, if any, of the variables which we utilize for analysis in our controller designs, such as force or velocity, are those sensed and monitored by humans. Many effects such as adaptation, learning, and attention shifts must also be considered in an analysis of the role of force feedback in a device for human use.

A discussion of psychophysical factors pertaining to haptic display of the dynamics of the piano action was undertaken in Chapter 2. Although pianists do not have independent control over intensity (loudness) and timbre (tone color) through the piano action, they do have independent control over the composite of these two coupled parameters and timing. It was suggested that one parameter might be *perceived* as another, and that this factor may be used to explain the piano's *perceptually* observed independent control of intensity and timbre and attendant unrestricted capabilities of musical expression. This hypothesis, if true, further motivates our goals of emulating the piano action's refined behavior in terms of hammer strike time and velocity and in terms of response forces using a keyboard-like haptic interface. After all, the response forces of the piano action are an integral component of the piano's behavior since the piano action and user constitute a coupled dynamical system. Interaction forces are part and parcel of that coupling and therefore come into play in the mapping from gesture to hammer strike parameters (or equivalently, from gesture to sound parameters). An analysis based on the coupling of two dynamical systems is consistent with the viewpoint that the human is neither a force nor motion source, but more appropriately modeled as a time-varying impedance. The ultimate response of this coupled system depends on the dynamical properties of both subsystems, human limb and piano action.

Another explanation of the disappointing utility of standard synthesizer keyboards at controlling musical sounds was offered in Chapter 2, based on the controls engineering concepts of *controllability* and *observability.* Although the selection of the sound parameters of the piano (intensity/timbre and timing) is *possible* from a simple synthesizer keyboard outfitted only with two-position switches, that selection is not as intuitive or 'organic' as in the case of the piano action. The sensitivity and predictability of these parameters to variations in the control input and the wealth of information about the fine effects of the control input which are hallmarks of the piano action are not available from the standard synthesizer keyboard. In the context of control by a human user, it can be said that the synthesizer keyboard does not support or encourage the development of various techniques as does the piano action. In particular, localized impedance variations or haptic features (localized in space, in time or in the parameter space of the input gesture) such as arise in the response of a dynamical system with changing kinematic constraints (including the piano action) are useful for increasing controllability and observability for a human user. Such localized features increase the store of available techniques and further, provide clues and suggestions for the development of such techniques which are consistent with common haptic experience in the physical world.

Dynamics of the Grand Piano Action

Second, with the aim of eventually emulating the piano action using human-in-the-loop simulation, we built a dynamical model of the action using the most numerically efficient model formulation —a formulation in independent coordinates. Armed with efficient modeling methods and accompanying computerized tools (Kane's method and *AUTOLEV*), we did not have to balk at the challenge of formulating constrained multibody dynamical models of the piano action. We didn't even hesitate to use constraint equations which are subject to change in order to eliminate dependent coordinates from the model, as has been the predilection of most researchers modeling variable structure systems. To encode the dynamics of changing kinematic constraints, we built models of the action in each of its constraint conditions and outfitted these models with the requisite indicator functions which would allow them to be linked together interactively during run-time. We extended the simulation methods currently available for simulation of such discontinuous systems to include a finite state machine. The finite state machine enables the accommodation of run-time dependence of submodel sequencing on user input. 'Artificial constraints' were introduced and used to cause those generalized coordinates which were not needed in a particular constraint configuration to nevertheless take on dependencies on other coordinates so that, when a time would arrive to exchange submodels in the simulator, the final conditions of that submodel could be passed on as initial conditions to the next, without risking violation of newly instated constraint equations. Details of the simulator software architecture were presented in Chapter 3, along with simulation results which verify that the effects of changing kinematic constraints on both the mapping from gesture to hammer strike parameters and the mechanical impedance of the piano action were successfully captured. 'Regulation' of the virtual piano action model is now possible with stop-button screw adjustments which fully parallel its referent, making fine tuning of behavior relatively easy —though probably still to be relegated to specialists. Piano technicians will certainly not be displaced with the rise of the virtual piano action.

Touchback Keyboard Design

Third, we built the haptic interface itself, which was featured in Chapter 4. An aluminum key, capstan-driven by a small high-performance motor through a 24:1 mechanical advantage, was designed to display, in the unpowered state, the impedance of a wooden piano key without the lead weights. Thus the controller driving the motor would be responsible for re-creating the dynamical effects of all elements of the piano action except the wooden key. The novel mechanical design of our keyboard which allows for the packing of all components into the tight space determined and enforced by the piano key spacing was documented. The piano action models built for simula*tion* in Chapter 3 were re-interpreted as controllers for the haptic interface, thus realizing a piano action simula*tor*. Sensed position is fed into the controller which responds with a torque to be displayed by the haptic interface. A parallel spring-damper pair is used to couple the physical hardware to the forward-dynamics simulation. This impedance-display implementation was contrasted to other possible implementations, including admittance-display.

Improved Controllers for the Virtual Wall

Certain destabilizing effects which tend to evoke non-physical behavior from simulated objects are particularly apparent with haptic display. Virtual objects with changing kinematic constraints are prone to contact instability, or chatter. Such chatter, being very uncharacteristic of real world objects will, upon encounter, immediately expunge any sense of virtual environment immersion which the user may have been enjoying. Contact instabilities and similar phenomena can be attributed to the fact that a virtual object's mechanical properties are simulated within a feedback controlled system in discrete time and mediated through a powered device. Virtual walls, for example, despite the fact that they are the simplest of objects containing a changing kinematic constraint, will exhibit contact instability (observed as sustained oscillations between contact and non-contact with the virtual wall) given certain parameter settings: high gains or low sample rates. Two insidious destabilizing effects are due to the zero order hold (a necessary element in any sampled data implementation) and what we have termed intersample threshold crossing (an artifact of the wall controller's switching nature and its sampled data implementation). Both of these effects can be quelled, however, using an assumed model of the full coupled system (which includes the human). Compensation for the effects of the zero order hold relies either on half-sample prediction or full state feedback using pole placement techniques from digital control theory and an assumed model of the driven key and the human limb. Compensation for the effects of intersample threshold crossing is performed with the use of the assumed model and an application of dead-beat control.

When no volitional control is involved, the literature shows that the human finger may be modeled as a second order linear impedance. Using the fact that the oscillatory behavior suffered by virtual walls is typically well above the frequency range of human voluntary movement, we have used static second order models for the human in the design of our new interactive controllers. This move is new to the literature. We think, however, that such methods can be used to great advantage and with ample justification. We are using local (short time duration) techniques to quell destabilizing effects which arise locally. Most importantly, such ideas hold much promise in their extension, such as when on-line system identification methods are used to keep the human models continually up to date, as further discussed below.

Stability Analysis of the new Virtual Wall Controllers

The new virtual wall controllers developed in Chapter 5 were analyzed for their usefulness in Chapter 6 with analytical treatments. Measures for the destabilizing effects of the zero order hold and intersample threshold crossing were sought for use in determining the worthiness for implementation of the new control techniques. An analytical solution for the damping coefficient which will balance the destabilizing effects of intersample threshold crossing under worst-case and non-existent bias force from the human was presented. Numerical extensions of this analytical result were made with a Poincaré map which was also derived, verified, and demonstrated.

7.2 Looking Forward

Psychophysical Investigations

With our seven-key touch-programmable keyboard in hand, we are ready to design and conduct a set of psychophysical experiments which investigate the utility of force feedback in musical expression. Since the synthesizer keyboard proved incapable of the range of musical expression found on the piano keyboard, and this fact seemed to be due to the lack of piano-like touch-response and the inappropriate substitution of velocity sensitivity for the piano's multifaceted mapping from gesture to sound parameters, we went about outfitting a synthesizer keyboard with force feedback and running interactive simulations of a piano action. Now we have an apparatus which can be programmed with various relationships between the feel at the key and the sound-response at the speaker ("soundboard"). With such an apparatus in hand, we may test whether a varying mechanical impedance such as that of the grand piano will help a musician develop and execute the fine control over musical sounds at the synthesizer keyboard which he already enjoys at the piano keyboard. Beyond the question *whether*, questions such as *how* may be asked. If we may learn how a human uses their sense of touch to develop manipulation strategies, then the very exiting possibility of designing instruments with maximized controllability and expressive potential will be opened up for exploration. I look forward to collaborative work in this area with experimental psychologists and experts in psychophysics and haptics.

Touchback Keyboard Development

The supporting hardware for this thesis, the seven-key Touchback Keyboard, also has a natural extension —its commercialization. Synthesizers constitute a very large and lucrative industry today, and there exists a sizable market for high-end synthesizer keyboards which feature optimum touch-response. The digital piano and synthesizer review-articles which appear at least annually in each of the electronic music magazines invariably claim that touch-response is the second most important factor for the buyer's consideration after sound quality. These reviews usually include tutorials on synthesizer actions and offer detailed critiques of how close the various presently available passive synthesizer actions come to duplicating the feel of the grand piano action.

Another testimonial to the viability of a touch-programmable keyboard-as-product is the interest with which each of CCRMA's industrial affiliate companies has been following this thesis work. Representatives from a number of synthesizer controller manufacturers, including Korg, Roland, Yamaha, and ZETA have been hearing about our work through CCRMA's industrial affiliates talks each year and have offered enthusiastic comments and support. This project is of course indebted to the industrial affiliate member companies since they have provided the financial support through the CCRMA affiliates program. Although the present design has yet to be optimized, made robust, and made truly cost effective, I regard the commercialization of force feedback in synthesizer keyboards as inevitable. Furthermore, force feedback will be the means to make concert-quality musical instruments available to the general public. Haptic interface technology promises gains in cost-effectiveness and manufacturability, whereas the existing hardware-intensive piano action design will only meet with increased costs in future times.

Modeling and Simulation Extensions

Our basic method for re-creating the touch response of a grand piano in a motorized keyboard is through human-in-the-loop simulation. Each key is coupled through a virtual spring and damper to a forward-dynamics simulation of a piano action model. The spring-damper coupling method is simple and provides for the filtering of the driving input and the response forces of the piano action. Although the spring-damper filter robs the displayed dynamics of high frequency components (crispness), especially when kinematic constraints change, it serves the very important purpose of suppressing the destabilizing effects of discrete simulation. However, the use of the more direct inverse dynamics simulation for impedance display (when force is computed directly from the virtual environment model, rather than with the use of a spring-damper coupler) is now of interest. The suppression or prevention of destabilizing effects would perhaps be more appropriately applied closer to the source of these numerical woes, in the discrete simulation algorithm itself.

For example, analytical treatments of numerical methods have produced restrictions on the stepsize which guarantee that the asymptotic behavior of the underlying differential equation be replicated when those numerical methods are used to simulate autonomous systems. These stepsize restrictions are a function of structural assumptions on the underlying differential equation and sometimes on initial conditions. The extension of these parameter bounds to non-autonomous system simulation such as is employed in human-in-the-loop schemes would assure the absence of numerical problems in inverse-dynamics simulations. Note: this idea is closely related to that discussed below under the heading 'extensions beyond the virtual wall'.

Virtual Wall Extensions

The appearance of destabilizing effects in an inverse dynamics setting was fully explored in the case of the virtual wall in Chapter 5. The virtual wall is a static system, implementable with a control law rather than with a numerical method. The virtual wall, because of its simplicity, allowed us to thoroughly explore a central theme of this thesis: Having chosen to use a discrete controller to endow a simple manipulandum with the dynamics of a more complex dynamical system, the simulation is prone to certain non-physical effects. When and if the full closed loop system (including the human) can be modeled, these non-physical effects can be eliminated using methods from digital control theory. Extensions to this theme center around the words *when and if* of the previous sentence. After all, an assumed model of the dynamics of the human will only be valid for very short time durations (for durations so short that volitional control will not be involved). While haptically exploring the rendered virtual object, the human's impedance properties will change. Two extensions to the approach of assuming a model for the human are immediately apparent.

- The model could be kept up to date at all times by incorporating an on-line system identification method into the controller. The force and motion signals at the contact point between human and manipulandum could be monitored and used, along with the known virtual environment dynamics, to ascertain the impedance properties of the human. The interaction controller (virtual environment display) would be continually adapted using this up-to-date human model to ensure always valid compensation for discrete implementation effects.

- The exact model for the human could be broadened to a model only satisfying certain structural restrictions. These structural restrictions would be chosen to lend themselves to analytical treatment and the generation of guidelines for the simulation method. Appropriately restrictive, yet not overly conservative assumptions about the set of behaviors which the human may be expected to exhibit within the system could be used to come up with haptic interface device and controller design guidelines which would ensure passive behavior of a simulated object. One such restriction which has been explored in the literature, especially by Colgate, has been the restriction of the human dynamics to passivity. This restriction, however, might be regarded as overly conservative within certain bandwidths and not conservative enough in others. Other sets within which to bound the human are worth considering. We would, after all, like to allow the expression of intentions by the user through occasional active behavior, yet still provide assurance that the power exchanges between user and interface device will be those which suggest interaction with real-world objects. The set of behaviors within which the human is restricted (for purposes of analysis) could be conformed to sets inside of which he may be expected to remain due to the action of natural and pre-existing constraints. This set would be larger in some respects than the set of all passive behaviors but smaller in other respects.

Finally, an extension to the virtual wall controllers presented in this thesis which would be very valuable from an implementation standpoint will be the incorporation of a velocity signal which is derived by numerical differentiation from a position signal in place of the direct velocity signal. Many haptic interface designs do not include a tachometer or other direct measure of velocity. Instead they rely on numerical differentiation of an encoder reading. Colgate's passivity analysis of the virtual wall supports the empirically observed destabilizing effects of large gains on the velocity term when a first-differenced position signal is used. The stability treatments in this thesis will be generalized in future work to cover this important effect.

Extensions beyond the Virtual Wall

The methods for the abolition of non-physical discrete simulation effects as they currently stand apply only to the simplest of virtual objects, the virtual wall. Similar factors, however, also underlie the non-physical behavior of dynamical, or multi-degree-of-freedom objects with changing constraints such as the virtual piano action. A major goal of future research will be to develop robust methods for simulating ***dynamical*** virtual objects which realistically support changing kinematic constraints. This will involve the design of new simulation algorithms which, like the analytical techniques and resulting design guidelines pertaining to the virtual wall, make use of assumptions about expected human behavior.

Integrated Haptic Interface Device and Simulator Design

Another attack on the problems associated with interactive dynamics with haptic display could be made with special attention to the design and utilization of computing hardware. The notion of computing hardware in the field of haptic interface of course encompasses the interface device itself in addition to the computer. Hardware architectures as suggested by certain simulator structures could be explored, such as parallel processing. Shared simulation across networks for multi-user environments also deserves attention. Interface design and software design go hand in hand for effective haptic display, and further gains are to be expected when the modeling procedure itself is factored into the structure of the simulator and computing hardware. Device and actuator design is another area which must be driven, especially with an approach which integrates controller design. We have, as proof of existence of room for improvement, the consummate example of effective manipulation and interactive behavior in the human system itself. The human hand and arm along with their controller are obviously very successful designs. Although the goals of haptic interface technology are re-create the mechanical properties of environments rather than to directly manipulate, exceptional human performance still stands as a testimony to the advantages of co-evolved and co-designed controller and actuator.

Bibliography

[1] Y. I. Alimov. 'On the application of Lyapunov's direct method to differential equations with ambiguous right sides.' *Avtomatika i telemekhanika*, 22, no. 7:817–830, July 1961.

[2] A. Askenfelt. 'Measuring the motion of the piano hammer during string contact.' Tech. Rep. STL-QPSR 4/1991, Royal Institute of Technology, Stockholm, 1991.

[3] A. Askenfelt and E. V. Jansson. 'From touch to string vibrations I. timing in the grand piano action.' *Journal of the Acoustical Society of America*, 88, no. 1:52–63, July 1990.

[4] A. Askenfelt and E. V. Jansson. 'On vibration sensation and finger touch in stringed instrument playing.' *Music Perception*, 9, no. 3:311–350, Spring 1992.

[5] R. P. Baker. 'Active touch keyboard.' United States Patent No. 4,899,631, Feb 1990.

[6] D. Baraff. 'Analytical methods for dynamic simulation of non-penetrating rigid bodies.' In: *Computer Graphics*, vol. 23-3, pp. 223–232. ACM, July 1989.

[7] D. Baraff. 'Fast contact force computation for nonpenetrating rigid bodies.' In: *Computer Graphics*, pp. 23–34. ACM, July 1994.

[8] G. Bartolini. 'Chattering phenomena in discontinuous control systems.' *International Journal of Systems Science*, 20, no. 12:2471–2481, 1989.

[9] R. Barzel. *Physically-Based Modeling for Computer Graphics : a Structured Approach*. Academic Press, Boston, 1992.

[10] R. Barzel and A. H. Barr. 'A modeling system based on dynamic constraints.' In: *Computer Graphics*, vol. 22 no. 4, pp. 179–88. ACM, Aug 1988.

[11] J. Baumgarte. 'Stabilization of constraints and integrals of motion in dynamical systems.' *Comput. Methods Appl. Mech. Engrg.*, 1:1–16, 1972.

[12] M. Bolas. Lecture, Design Forum, Stanford University, May 1994.

[13] X. Boutillon. 'Model for piano hammers: experimental determination and digital simulation.' *Journal of the Acoustical Society of America*, 83, no. 2:746–754, February 1988.

[14] R. Brockett. 'Language driven hybrid systems.' In: *33rd IEEE Conference on Decision and Control*, vol. 4, pp. 4210–4214, 1994.

[15] C. Budd and F. Dux. 'Chattering and related behaviour in impact oscillators.' Tech. Rep. AM-93-02, University of Bristol, 1992.

[16] M. Bühler and D. E. Koditscheck. 'From stable to chaotic juggling: theory, simulation, and experiments.' In: *International Conference on Robotics and Automation*, vol. 3, pp. 1976–81. IEEE, May 1990.

[17] M. Bühler, D. E. Koditscheck, *et al.*. 'A family of robot control strategies for intermittent dynamical environments.' In: *International Conference on Robotics and Automation*, vol. 3, pp. 1296–1301. IEEE, May 1989.

[18] C. Cadoz, L. Lisowski, *et al.*. 'A modular feedback keyboard design.' *Computer Music Journal*, 14, no. 2:47–51, Summer 1990.

[19] C. Cadoz, A. Luciani, *et al.*. 'Responsive input devices and sound synthesis by simulation of instrumental mechanisms: the cordis system.' *Computer Music Journal*, 8, no. 3:60–73, Fall 1984.

[20] C. Cadoz, A. Luciani, *et al.*. 'Cordis-anima: a modeling and simulation system for sound and image synthesis —the general formalism.' *Computer Music Journal*, 17, no. 1:19–29, Spring 1993.

[21] J. E. Colgate. 'Coordinate transformations and logical operations for minimizing conservativeness in coupled stability criteria.' *Journal of Dynamic Systems, Measurement and Control*, 116, no. 4:643–649, December 1994.

[22] J. E. Colgate and J. M. Brown. 'Factors affecting the z-range of a haptic display.' In: *International Conference on Robotics and Automation*, 1994.

[23] J. E. Colgate, P. E. Grafing, *et al.*. 'Implementation of stiff virtual walls in force-reflecting interfaces.' In: *Virtual Reality Annual International Symposium*, pp. 202–208, Seattle,WA, September 1993. IEEE.

[24] J. E. Colgate and G. Schenkel. 'Passivity of a class of sampled-data systems: application to haptic interfaces.' In: *American Control Conference*, pp. 3236–3240. IEEE, June-July 1994.

[25] J. E. Colgate, M. C. Stanley, *et al.*. 'Issues in the haptic display of tool use.' In: *IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, pp. 140–145, 1995.

[26] M. R. Cutkosky. *Brief Lessons in High Technology*, chap. Robotics: a dream as old as antiquity, pp. 155–189. Portable Stanford Book Series. Login Publishers Consortium, 1985.

[27] P. Dijksterhuis. *De Piano*. Nederlands Akoestisch Genootschap, 1965.

[28] S. Djerassi. 'Imposition of constraints.' *Transactions of the ASME*, 61:434–439, 1994.

[29] S. D. Eppinger and W. P. Seering. 'On dynamic models of robot force control.' pp. 29–34, 1986.

[30] J.-L. Florens, A. Razafindrakoto, *et al.*. 'Optimized real time simulation of objects for musical synthesis and animated image synthesis.' In: *International Computer Music Conference*. ICMA, 1986.

[31] G. F. Franklin, J. D. Powell, *et al.*. *Digital Control of Dynamic Systems*. Addison Wesley, 2nd edn., 1990.

[32] J. Garcia de Jalón and E. Bayo. *Kinematic and Dynamic Simulation of Multibody Systems: the Real-Time Challenge*. Springer-Verlag, 1994.

[33] B. Gillespie. 'The virtual piano action: design and implementation.' In: *Proceedings of the International Computer Music Conference, Held: Aahus, Denmark*, pp. 167–170. ICMA, 1994.

[34] B. Gillespie and L. Rosenberg. 'Design of high-fidelity haptic display for one-dimensional force reflection applications.' In: *Telemanipulator and Telepresence Technology, Proceedings of the International Society of Optical Engineers East Coast Conference, Held: Boston, MA*, pp. 44–54. SPIE, 1994.

[35] B. J. Gilmore and R. J. Cipra. 'Simulation of planar dynamic mechanical systems with changing topologies: Part 1–characterization and prediction of the kinematic constraint changes.'

[36] B. J. Gilmore and R. J. Cipra. 'Simulation of planar dynamic mechanical systems with changing topologies: Part ii – implementation strategy and simulation results for example dynamic systems.'

[37] J. Guckenheimer. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. Springer-Verlag, 1983.

[38] J. K. Hahn. 'Realistic animation of rigid bodies.' *Computer Graphics*, 22, no. 4:299–308, August 1988.

[39] A. Z. Hajian and R. D. Howe. 'Identification of the mechanical impedance of human fingers.' In: *Dynamic Systems and Controls*, vol. 1 no. 55, 1994.

[40] D. E. Hall. 'Piano string excitation V: Spectra for real hammers and strings.' *Journal of the Acoustical Society of America*, 83, no. 4:1627–1638, April 1988.

[41] B. Hannaford and R. Anderson. 'Experimental and simulation studies of hard contact in force reflecting teleoperation.' In: *International Conference on Robotics and Automation*, vol. 1, pp. 584–589. IEEE, April 1988.

[42] M. W. Hart, C. H. Fuller and W. S. Lusby. 'A precision study of piano touch and tone.' *Journal of the Acoustical Society of America*, 6:80–94, 1934.

[43] E. J. Haug and R. C. Deyo. *NATO Advanced Research Workshop on Real-Time Integration Methods for Mechanical System Simulation (1989 : Snowbird, Utah)*. Springer-Verlag, 1991.

[44] E. J. Haug, S. C. Wu, *et al.*. 'Dynamics of mechanical systems with coulomb friction, stiction, impact and constraint addition-deletion –i.' *Mechanism and Machine Theory*, 21, no. 5:401–406, 1986.

[45] J. N. Helmann. *The Consciously Controlled Piano Tone : Most Natural Approach to the Problem of Artistic Piano Playing*. J. N. Helmann, 1969. Translated from the original Russian by Luba Helmann Berton. Year: 1950.

[46] N. Hogan. 'Stable execution of contact tasks using impedance control.' In: *International Conference on Robotics and Automation*, pp. 1047–1053. IEEE, 1987.

[47] P. J. Holmes. 'The dynamics of repeated impacts with a sinusoidally vibrating table.' *Journal of Sound and Vibration*, 84, no. 2:173–189, 1981.

[48] R. M. Howe. 'Techniques for optimizing computer performance in real-time flight simulation.'

[49] J. Hyde, M. Tremblay, *et al.*. 'An object-oriented framework for event-driven dextrous manipulation.' In: *Proceedings, 4th International Symposium on Experimental Robotics, Stanford, CA*, June 1995.

[50] J. M. Hyde and M. R. Cutkosky. 'Contact transition control: an experimental study.' In: *International Conference on Robotics and Automation*, vol. 1, pp. 363–368. IEEE, May 1993.

[51] P. M. Isaaks and M. F. Cohen. 'Controlling dynamic simulation with kinematic constraints, behavior functions and inverse dynamics.' *Computer Graphics*, 21, no. 4:215–224, July 1987.

[52] S. J. Jeans. 'Lecture before the english music teachers' association, London.' Reprinted New York Times, Jan 8, 9 1939.

[53] W. Kaempfert. 'This week in science.' New York Times Column, January 15 1939.

[54] T. R. Kane. *Dynamics, Theory and Applications*. McGraw-Hill, New York, 1985.

[55] D. C. Karnopp and R. C. Rosenberg. *Introduction to Physical System Dynamics*. McGraw-Hill, 1975.

[56] Y. Katayma, Y. Hanjo, *et al.*. 'Event-driven motion-module switching mechanism for robot motion control: concept and experiment.' In: *Dynamic Systems and Controls*, vol. 55-1, pp. 199–206, 1994.

[57] P. U. Lee, D. C. Ruspini, *et al.*. 'Dynamic simulation of interactive robotic environment.' In: *International Conference on Robotics and Automation*, vol. 2, pp. 1147–1152. IEEE, May 1994.

[58] T. Letowski. 'Timbre, tone color, and sound quality: concepts and definitions.' *Archives of Acoustics*, 17, no. 1:17–30, 1992.

[59] K. C. Lin and R. M. Howe. 'A new approach to the real-time simulation of control systems with discontinuities.' In: *American Control Conference*, vol. 2, pp. 2122–2127. American Autom. Control Council, June 1991.

[60] S.-T. Lin and H. K. Yae. 'Controller design for robot tasks involving free and contact motion.' In: *American Control Conference*, vol. 1, pp. 765–770, June 1992.

[61] P. Lötstedt. 'Mechanical systems of rigid bodies subject to unilateral constraints.' *SIAM Journal of Applied Math.*, 42, no. 2:281–296, April 1982.

[62] P. Lötstedt. 'Numerial simulation of time-dependent contact and friction problems in rigid body mechanics.' *SIAM Journal of Sci. Stat. Comput.*, 5, no. 2:370–393, June 1984.

[63] A. Luciani, S. Jimenez, *et al.*. 'Computational physics: a modeler-simulator for animated physical objects.' In: *International Computer Music Conference*. ICMA, 1991.

[64] K. E. MacLean and W. K. Durfee. 'An apparatus to study the emulation of haptic feedback.' In: *Dynamic Systems and Controls*, vol. 57-2, pp. 615–621, 1995.

[65] T. H. Massie and J. K. Salisbury. 'Probing virtual objects with the phantom haptic interface.' In: *Dynamic Systems and Controls*, Chicago, Illinois, November 1994. ASME.

[66] D. McRuer and E. S. Krendel. 'Mathematical models of human pilot behavior.' *AGARDograph 188*, January 1974.

[67] J. K. Mills. 'Stability of robotic manipulators during transition to and from compliant motion.' *Automatica*, 26, no. 5:861–874, 1990.

[68] J. K. Mills and A. A. Goldenberg. 'Force and position control of manipulators during constrained motion tasks.' *IEEE Transactions on Robotics and Automation*, 5, no. 1:30–46, February 1989.

[69] J. K. Mills and D. M. Lockhorst. 'Control of robotic manipulators during general task execution: a discontinuous control approach.' *The International Journal of Robotics Research*, 12, no. 2:146–163, April 1993.

[70] M. Minsky. *Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display*. Ph.D. thesis, Massachusetts Institute of Technology, 1995.

[71] M. Minsky, O.-Y. Ming, *et al.*. 'Feeling and seeing: issues in force display.' In: *Computer Graphics*, vol. 24-2, pp. 235–243. ACM Transactions on Computer-Human Interactions, March 1990.

[72] B. Mirtich. 'Hybrid simulation: Combining constraints and impulses.' In: *Proceedings of First Workshop on Simulation and Interaction in Virtual Environments*, July 1995.

[73] B. Mirtich and J. Canny. 'Impulse-based dynamic simulation.' In: *Proceedings of Workshop on Algorithmic Foundations of Robotics*, February 1994.

[74] B. Mirtich and J. Canny. 'Impulse-based simulation of rigid bodies.' In: *Proceedings of 1995 Symposium on Interactive 3D Graphics*, April 1995.

[75] D. Mitra. 'The absolute stability of high-order discrete-time systems utilizing the saturation non-linearity.' *IEEE Trans. Circuits and Systems*, 25, no. 6:365–371, 1978.

[76] M. Moore and J. Wilhelms. 'Collision detection and response for computer animation.' *Computer Graphics*, 22, no. 4:289–298, August 1988.

[77] C. P. Neuman and V. D. Tourassis. 'Discrete dynamic robot models.' *IEEE Transactions on Systems, Man, and Cybernetics*, 15, no. 2:193–204, March/April 1985.

[78] O. Ortmann. *The Physical Basis of Piano Touch and Tone.* Dutton, New York, 1925.

[79] W. Pfeiffer. *Über Dampfer, Federn und Spielart.* Verlag Das Musikinstrument, 196-? in German.

[80] W. Pfeiffer. *The Piano Hammer.* Verlag Das Musikinstrument, 1967.

[81] W. Pfeiffer. *The Piano Key and Whippen.* Verlag Das Musikinstrument, 1967.

[82] J. C. Platt and A. H. Barr. 'Constraint methods for flexible models.' *Computer Graphics*, 22, no. 4:279–287, August 1988.

[83] M. Raibert. 'Dynamic legged robots.' In: R. Jarvis, ed., *Proceedings of the International Symposium and Exposition on Robots, 19th ISIR*, pp. 86–93, 1994.

[84] A. A. Reblitz. *Piano Servicing, Tuning, and Rebuilding.* The Vestal Press, 1976.

[85] A. Riddell. 'A meta-action for the grand piano.' In: *International Computer Music Conference*, pp. 395–397, Glasgow, 1990.

[86] L. B. Rosenberg. *Virtual fixtures: perceptual overlays enhance operator performance in telepresence tasks.* Ph.D. thesis, Stanford University, 1994.

[87] E. Roxin. 'On generalized dynamical systems defined by contingent equations.' *Journal of differential equations*, 1:188–205, 1965.

[88] E. Roxin. 'Stability in general control systems.' *Journal of differential equations*, 1:115–150, 1965.

[89] D. B. Schaechter and D. A. Levinson. 'Interactive computerized symbolic dynamics for the dynamicist.' *Journal of Astronautical Sciences*, 36, no. 4:365–388, Nov-Dec 1988.

[90] S. A. Schneider and R. H. Cannon Jr. 'Object impedance control for cooperative manipulation: theory and experimental results.' *IEEE Journal of Robotics and Automation*, 8, no. 3:383–394, June 1992.

[91] C. E. Seashore. *Psychology of Music.* McGraw-Hill, New York, 1938. also in Dover, 1967.

[92] J. O. Smith III. 'Physical modeling using digital waveguides.' *Computer Music Journal*, 16, no. 4:74–87, Winter 1992.

[93] T. N. Topper and B. L. Wills. 'The computer simulation of piano mechanisms.' *International Journal of Modelling and Simulation*, 7, no. 4:135–139, 1990.

[94] M. R. Tremblay and M. R. Cutkosky. 'Using sensor fusion and contextual information to perform event detection during a phase-based manipulation task.' In: *Proceedings 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, vol. 3, pp. 262–267, May 1995.

[95] J. C. Tsai and J. E. Colgate. 'Stability of discrete time systems with unilateral nonlinearities.' In: *Dynamic Systems and Controls*, vol. 57-2. ASME, 1995.

[96] V. I. Utkin. 'Discontinuous control system: state of art in theory and applications.' In: *Automatic Control - World Congress, 1987*, vol. 1 of *Selected papers from the 10th Trieennail World Congress*, pp. 25–44. IFAC, 1987.

[97] G. Van den Berghe. 'Modellering van een pianotoetsmechaniek.' Tech. Rep. UDC:681.51(043), Faculteit Der Toegepaste Wetenschappen, Departement Elektrotechniek, Katholieke Universiteit Leuven, 1992.

[98] G. Van den Berghe, B. De Moor, *et al.*. 'Modeling a grand piano key action.' *Computer Music Journal*, 19, no. 2:15–22, Summer 1995.

[99] S. Van Duyne, J. R. Pierce, *et al.*. 'Traveling wave implementation of a lossless mode-coupling filter and the wave digital hammer.' In: *Proceedings, International Computer Music Conference, Aarhus*, 1994.

[100] S. A. Van Duyne, J. R. Pierce, *et al.*. 'Traveling wave implementation of a lossless mode-coupling filter and the wave digital hammer.' In: *International Computer Music Conference*, 1994.

[101] J. T. Wang and R. L. Huston. 'Kane's equations with undetermined multipliers–application to constrained multibody systems.' *Transactions of the ASME*, 54:424–429, June 1987.

[102] Y. Wang. 'Dyanamic analysis of mechanical systems with time-varying topologies: part 1, dynamic model and stability analysis.'

[103] R. A. Wehage and E. J. Haug. 'Dynamic analysis of mechanical systems with intermittent motion.' *Transactions of the ASME*, 104:778–784, October 1982.

[104] B. W. White, F. A. Saunders, *et al.*. 'Seeing with the skin.' *Perception and Psychophysics*, 7:23–27, 1970.

[105] S. C. Wu, Y. S. M., *et al.*. 'Dynamics of mechanical systems with coulomb friction, stiction, impact and constraint addition-deletion –iii.' *Mechanism and Machine Theory*, 21, no. 5:417–425, 1986.

[106] S. C. Wu, S. M. Yang, *et al.*. 'Dynamics of mechanical systems with coulomb friction, stiction, impact and constraint addition-deletion –ii.' *Mechanism and Machine Theory*, 21, no. 5:407–416, 1986.

[107] Y. Xu, J. M. Hollerbach, *et al.*. 'A nonlinear PD controller for force and contact transient control.' *IEEE Control Systems Magazine*, 15, no. 1:15–21, February 1995.

[108] K. Yamamoto. Personal Communcation, May 1995.

[109] J. Yen, E. J. Haug, *et al.*. 'Numerical methods for constrained equations of motion in mechanical system dynamics.' *Mechanics of Structures and Machines*, 19, no. 1:41–76, 1991.

[110] S. Yoon, R. M. Howe, *et al.*. 'Constraint violation stabilization using gradient feedback in constrained dynamics simulation.' *Journal of Guidance, Control and Dynamics*, 15, no. 6:1467–1474, November-December 1992.