

Velocity-driven Audio-Haptic Interaction With Real-Time Digital Acoustic Models

Stephen Sinclair



Input Devices and Music Interaction Laboratory
Schulich School of Music
McGill University
Montreal, Canada

August 14, 2012

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

© 2012 Stephen Sinclair

Abstract

Acoustic models driven by real-time velocity signals can suffer unduly from quality issues due to sampling and differentiation, especially at high sampling rates. In audio-haptic friction interaction, as found in a bowed string simulation for example, this noise appears as a gritty or dry feel, and is audible in the sound.

In this thesis, two approaches to this problem are proposed: firstly, reduction of the sensitivity of the model to velocity noise by the application of a position-dependent friction model; secondly, the improvement of velocity estimation by means of filtering and enhanced sensing.

Several estimators are compared, by means of parameter optimisation, to direct velocity measurement in order to find a good trade-off between filter-imposed delay and noise rejection. Optimised estimators are then compared by subjects in an online scenario to test their respective effect on the impedance range and noise qualities of a bowed string friction display.

Resumé

Les modèles acoustiques commandés en temps réel par des signaux de vitesse peuvent souffrir indûment de problèmes de qualité dus à l'échantillonnage et à la différenciation de la position, en particulier à des taux d'échantillonnage élevé.

Dans l'interaction audio-haptique basée sur la friction, que l'on trouve dans la simulation d'une corde frottée par exemple, ce bruit se présente comme une sensation granuleuse ou sèche, et est audible dans le son.

Nous proposons dans cette thèse deux solutions à ce problème: d'une part, la réduction de la sensibilité au bruit du modèle de simulation par l'application d'un modèle de friction dépendant de la position; d'autre part, l'amélioration de l'estimation de la vitesse en utilisant différents filtres ainsi que des capteurs supplémentaires.

Plusieurs estimateurs sont comparées entre eux et à la mesure directe de la vitesse, dans le but de trouver un compromis acceptable entre le retard imposé par le filtrage et la réduction du bruit.

Ces estimateurs optimisés sont ensuite évalués par des participants afin de tester leurs effets sur la gamme d'impédance obtenue ainsi que sur la qualité du bruit dans la simulation de corde frottée.

Acknowledgments

As I've come to learn, a principle difference between a Master's and a Ph.D thesis is that in the latter there are many more people to thank.

This work would certainly not have been possible if it were not for many contributions. The Input Devices and Music Interaction Laboratory (IDMIL) at McGill's Schulich School of Music, and its director, my supervisor, Marcelo M. Wanderley, have been a source of constant feedback and encouragement throughout this process. My fellow students participated greatly in my work, both for discussion, ideas, and solutions. In particular I'd like to acknowledge Avrum Hollinger, Joseph Malloch, Mark Zadel, and Marcello Giordano, colleagues whose knowledge, ideas and skills are an inspiration. I must also thank the surprisingly willing participants in my small experimental studies, which had to be repeated as new data was found—for this I am ever grateful.

Collaborations within McGill and with other institutions have led to turning points in this work. Instruction from my co-supervisor, Vincent Hayward, formerly CIM, McGill, now at ISIR, UPMC, on haptics and computer-controlled systems has become a cornerstone of my expertise in this area. Consultation with Jean-Loup Florens at ACROE, INPG, as well as former ACROE students Olivier Tache and Julien Castet, gave me significant experience with the TGR, along with memorable discussions on haptics. My understanding of bowed string physics would have amounted to nil without Gary Scavone in McGill Music Technology, who reviewed and clarified many concepts for me, and who diligently maintains the STK software I worked from. A visit abroad to work with Sile O'Modhrain, formerly SARC, Queens U. Belfast, now at U. Michigan, was equally inspirational.

No less important, and preceding my enrolment, I am lucky to have had early hands-on experience working closely with my father, Ian Sinclair, at MPBT on the Freedom 6S haptic master. Working professionally on such a finely-tuned instrument with my Dad is an experience I would never trade. It has always been the primary factor in inspiring me to pursue force feedback interaction as a topic of research.

I also owe a huge debt of gratitude to Shabana, to both my parents, and to my friends and family for unending encouragement, support, and patience.

Finally, financial support from the Natural Sciences and Research Council of Canada (NSERC), MPB Technologies Inc., and the Center Interdisciplinary Research in Music Media and Technology (CIRMMT) was of course indispensable to my success.

Contents

1	Introduction	1
1.1	Overview	1
1.1.1	Motivation	1
1.2	Problem domain	3
1.2.1	Haptic and force feedback interaction	3
1.2.2	Acoustic haptic interaction	4
1.2.3	Audio-haptic interaction	5
1.3	Bowed string modeling and synthesis	9
1.3.1	String model	10
1.3.2	Efficiency and trade-offs	12
1.3.3	Frictive excitation mechanism	13
1.3.4	The digital waveguide bowed string	14
1.4	Force feedback from the bowed string	17
1.4.1	Noise problems in acoustic friction display	17
1.5	Structure of the thesis	18
2	Background	20
2.1	Requirements of haptic display	20
2.2	Passive tool display	22
2.2.1	Use in bowed string interaction	23
2.3	Influence of signal quality on haptic interaction	25
2.4	Approaches to velocity-based interaction	29
2.4.1	Passive actuator approaches	29
2.4.2	Electrical approaches	31

2.4.3	Algorithmic approaches	31
2.5	Conclusion	33
3	Influence of signal quality on friction display	34
3.1	Distortion caused by two types of signal error	36
3.1.1	Friction measurements	37
3.1.2	Noise	39
3.1.3	Delay	41
3.2	Experiment #1: noisy and delayed friction display	43
3.2.1	Hypothesis: Velocity-error distortion regimes	43
3.2.2	Materials and configuration	45
3.2.3	Methodology	46
3.2.4	Results	47
3.2.5	Discussion	49
3.3	Virtual tool display	51
3.3.1	Choice of coupling coefficients	51
3.4	Application to bowing—representation of the bow and bow hair	58
3.4.1	Expected friction force in real bowing	58
3.4.2	The bow hair	59
3.4.3	Inertia	62
3.4.4	Other considerations	63
3.5	Discussion	63
4	Improvement by reduced dependence on velocity	65
4.1	Position-constraint friction	66
4.2	The TWO-POINT model	67
4.2.1	Discussion	68
4.3	The DISTPLUCK model	69
4.3.1	Partial slip provides anti-aliasing	72
4.3.2	Offline simulations	73
4.3.3	Online simulations	78
4.3.4	Conclusion	79

5	Improvement by measurement and estimation	82
5.1	Velocity-coupled bowed string	83
5.1.1	Bow force from the digital waveguide	84
5.1.2	Correspondance to the implicit junction	86
5.2	Sensors	89
5.2.1	Position	89
5.2.2	Velocity	90
5.2.3	Acceleration	92
5.3	Differentiators	93
5.3.1	Backward difference	93
5.3.2	Least squares fit	94
5.3.3	Adaptive windowing	96
5.3.4	Levant's differentiator	97
5.4	Fusion techniques	99
5.4.1	Complementary filter	100
5.4.2	Kalman filter	100
5.4.3	Hybrid solutions	103
5.5	Trade offs	104
5.5.1	Computational requirements	106
5.5.2	A priori knowledge	107
5.6	Conclusion	108
6	Optimisation and Numerical Evaluation	110
6.1	Signal recordings	110
6.1.1	Sensor noise characterisation	112
6.2	An error metric for signal comparison	112
6.2.1	Error metric validation	115
6.2.2	Comparing velocity signals	116
6.3	Optimization	117
6.3.1	Error surface calculation details	117
6.3.2	Multi-objective global optimisation	118
6.4	Results	123
6.4.1	Parameter selection	123

6.4.2	Performance	129
6.4.3	Validation	132
6.4.4	Execution time	133
6.4.5	Discussion	133
6.4.6	Gain-specific optimisation	136
6.5	Discussion	136
6.6	Conclusion	141
7	Online Evaluation	142
7.1	Experiment #2	143
7.1.1	Stimulus	144
7.1.2	Conditions	145
7.1.3	Apparatus and Methodology	146
7.2	Results	146
7.2.1	Maximum impedance judgement	147
7.2.2	Subjective noisiness ratings	148
7.3	Further investigation of perceived noise	149
7.4	Discussion	149
7.5	Conclusion	152
8	Discussion	154
8.1	Generalizability of results	154
8.1.1	Generalizability of DISTPLUCK	154
8.1.2	Applicability of velocity estimation improvements	156
8.2	Observations	157
8.2.1	Perception of noise in the presence of an acoustic model	157
8.2.2	On device-in-the-loop acoustic interaction	158
8.3	Conclusion	160
9	Conclusion	161
9.1	Review	162
9.2	Contributions	162
9.2.1	Recommendations for hardware design	162
9.2.2	Signal processing	163

9.2.3	Haptic simulation	164
9.3	Future research	164
9.4	Concluding remarks	167
A	Equipment	168
A.1	The Ergon-X system	168
A.2	Calibration	169
A.3	Friction characteristics	171
B	Parameter tuning	175
B.1	Automatic optimisation challenges	175
B.2	Parameter selection with respect to error and delay surface inspection . . .	176
B.3	Position-based estimators	176
B.3.1	Low-pass filter	176
B.3.2	Least-squares FIR filter	177
B.3.3	First-order adaptive window best-fit	178
B.3.4	Levant's differentiator	181
B.3.5	2 nd -order Kalman filter	183
B.4	Fusion estimators incorporating accelerometer measurement	184
B.4.1	Complementary filter	185
B.4.2	3 rd -order Kalman filter	186

List of Figures

1.1	Rosin friction: standard curve vs. a hysteretical relation	13
1.2	Block diagram of the digital waveguide bowed string	15
1.3	The transmission coefficient for several values of vertical bow force	16
2.1	Bowed string force feedback interaction model from Florens	23
2.2	Interaction model proposed by Berdahl	24
3.1	Coulomb friction with and without a linearly damped section	39
3.2	Velocity and force for a friction display with added noise	40
3.3	Velocity and force for a friction display with added delay	42
3.4	A possible subdivision of different distortion regimes for noise-error and delay-error during friction interaction.	44
3.5	A subject performing Experiment #1.	46
3.6	Discrimination and stability thresholds determined from experiment 1 . . .	48
3.7	An intermediate mass modeling a haptic tool	52
3.8	The RMS error of observed velocity for the mass-spring-damper tool model during steady state, as a function of stiffness and damping coefficients . . .	55
3.9	Bode plots of double-differentiated frequency response for a inertial system which modifies the apparent mass of the end effector	57
3.10	Models for the role of bow hair in a virtual bow coupling	61
4.1	Movement of the anchor point in a position-dependent friction model . . .	66
4.2	Mechanical interpretation of virtual points used in the TWO-POINT model .	67
4.3	String velocity and resonator input signal in DISTPLUCK	71
4.4	The DISTPLUCK model with digital waveguide resonator	72

4.5	Curves denoted by different versions of $\alpha(z)$ in DISTPLUCK	74
4.6	Spectra for the velocity pulses and resulting string velocity during the steady state signal in two configurations of the DISTPLUCK model	75
4.7	String velocity, v_s , in noisy and noiseless conditions for BOWED, TWOPOINT, and (c) DISTPLUCK models	76
4.8	String velocity during the attack and steady-state of each model for clean and noise conditions	77
4.9	Two zoom levels of a recording of DISTPLUCK running at 30 kHz on a haptic device	80
5.1	Bowed string digital waveguide, enhanced with a force output to provide haptic feedback	84
5.2	Bow impedance and transmission coefficient for several values of vertical bow pressure	86
5.3	The TGR device in a horizontal 1-DOF configuration, coupled to the Trans-Tek Series 100 LVT tachometer.	91
5.4	Demonstrating the usage of non-linear differentiators to help correct accelerometer bias	105
6.1	A sample of the recorded signals used for optimisation captured simultaneously with three sensors	111
6.2	The noise distribution under the experimental conditions while the device was held still	113
6.3	Demonstration of time-adjusted signal comparison method on a damped sinusoid	114
6.4	Delay detection and subsequent RMS error evaluation applied to each recording for error metric validation	115
6.5	Error evaluation and parameter optimization process.	117
6.6	Maximum and RMS acceleration vs. friction for each recording as measured by the accelerometer.	127
6.7	Examples of Levant's differentiator behaviour	127
6.8	Error and delay evaluation for each recording using optimised parameter sets for each estimator	130

6.9	Error and delay evaluation for each recording using optimised parameter sets for each estimator	130
6.10	Error and delay evaluation for each recording using optimised parameter sets for each estimator, non-parametric box plot form	131
6.11	Error and delay evaluation, significance of differences according to Wilcoxon rank-sum test	132
6.12	Comparison of execution times for each estimator	134
6.13	All estimators applied to a section of a bow stroke in recording $F_{\max} = 1.92$ N	135
6.14	Error and delay evaluation for low-gain recordings, with significance of differences according to Wilcoxon rank-sum test	137
6.15	Error and delay evaluation for high-gain recordings, with significance of differences according to Wilcoxon rank-sum test	138
6.16	3 rd -order Kalman filter estimators applied to the attack of a bow stroke . .	139
6.17	Example comparisons of velocity and accelerometer data	140
7.1	Maximum friction force for bowed string interaction	147
7.2	Subjective ratings of noisiness for several estimators	148
7.3	Spectra of stationary noise for several estimators	150
7.4	Probability densities of stationary noise for several estimators	151
8.1	Two close-ups of the attack of the TWO-POINT model of similar gestures with and without bow-string friction feedback	159
A.1	The TGR device with a bow attached	170
A.2	Device calibration setup	171
A.3	Force-velocity plots of sinusoidal signals showing friction behaviour	172
B.1	Error and delay surfaces for a 2 nd -order Butterworth low-pass filter applied to each recording	177
B.2	Error and delay surfaces for the least squares fit FIR estimator applied to each recording	178
B.3	A two-dimensional error surface for one recording of the FOAW best-fit estimator demonstrating the separability of the parameters	179

B.4	The 1-D cross-section of the FOAW error surface along the window size axis for error and delay	179
B.5	Error and delay surfaces for Levant's 2-sliding observer differentiator applied to each recording	181
B.6	Error and delay surfaces for Levant's 2-sliding observer differentiator, post-filtered with a BW2 low-pass filter, applied to each recording	182
B.7	Example of the 2-D error surface for the 2 nd -order Kalman estimator . . .	183
B.8	One-dimensional error and delay surfaces for the 2 nd -order Kalman estimator applied to each recording	184
B.9	Error and delay surfaces for the complementary filter estimator with varying parameters for BW2 low- and high-pass filter cut-off frequencies	185
B.10	Two-dimensional view of the error and delay surfaces for the 3 rd -order Kalman estimator with varying parameters for process and accelerometer covariance	186

List of Tables

3.1	Common bow weights	62
6.1	Complete list of estimators tested	124
6.2	Final parameters for each estimator based on global optimisation across all recordings	125
A.1	Properties of the Ergon-X system	169
A.2	Friction characteristics as reported by least squares fit	173

Chapter 1

Introduction

1.1 Overview

Haptic force feedback, defined herein as the application of grounded hand-held robotic interfaces to virtual reality simulation and interaction, can provide a sense of presence and embodiment of virtual objects that is otherwise difficult to achieve.

It has seen use in scientific applications such as molecular docking and nanotechnology, design applications such as industrial prototyping, medical applications such as telesurgery and simulation, and artistic applications such as 3D modeling [18].

In this dissertation, we are concerned with applying this technology to real-time musical interaction. We will tackle issues in the use of force feedback for providing haptic cues related to live interaction with sound-producing virtual objects. In particular we will focus on velocity-based, i.e. frictive, or *bowing* interaction with digitally-modeled resonators.

1.1.1 Motivation

In discussing this work with others, a question that has often occurred is, *why simulate interaction with musical objects, when one could do much better using a real musical instrument?* Certainly, an audio-enabled haptic simulation—a touchable virtual environment accompanied by sound—could be used for music making, but in the current state of the art it would make a very limited instrument indeed.

However, this fact is a large part of the motivation for research in this area. It makes us pose a useful question: *what*, then, is lacking? What are the limitations of haptic interaction

that leave something to be desired, that would force us to prefer real instruments? Haptics, and indeed virtual reality in general, are still young technologies. We do not ask at this point whether it offers a completely convincing experience, but rather we must ask what is still missing to present the user with a fully embodied environment.

There are many limitations to acknowledge: limited workspace, limited points of interaction, limitations to tactile display, stability concerns, impedance limitations, etc. Although improvements in all these areas are warranted, before blindly proceeding it is useful to characterise their impact.

It also brings up questions of a larger scope, such as, *what makes a good instrument?* and *how do we perceive one?* These questions are, of course, extremely broad, since one must consider not only the physical and perceptual aspects of human interaction, but also the cultural history and social milieu which encompasses our concept of musical instrument. Nonetheless, in today's age of digital music-making on laptops, touch-screen tablets, and hands-free interfaces, the role of touch perception in music seems to be doubly poignant. Virtual haptic display is one tool at our disposal that can help investigate such questions.

Conversely, we consider audio-haptic simulation of musical instruments as representative of some still-difficult areas in the study and design of real-time simulation and virtual reality. Trained musicians have spent a considerable amount of time throughout their lives, often from a very young age, intimately coupled to an acoustically vibrating object, and as a result have developed a level of familiarity with their tool often considered unsurpassed by other disciplines. Therefore, music may provide a unique vector for evaluation of interaction with acoustic simulations.

Additionally, although in this work we consider only musical interaction in traditional bowing and frictive actions, this technology could in the future provide a means for studying alternative interactive paradigms that would be difficult to achieve in the real world, or simply more convenient to prototype on the computer. For instance, it could play a role in designing musical interactions with simulated materials in various geometric configurations, as created in acoustic physical modeling software such as Modalys [39].

Finally, as motion capture and haptic simulation increases its role in artistic areas such as the demand for 3D puppetry in film, television, and video games [64, 68], we imagine that it is likely this trend will eventually be accompanied by a demand for corresponding sound synthesis, as has happened with trends in computer animation [83]. Thus there may be applications of this technology in the media industry, in the form of virtual foley

[129]. Currently this is limited for commercial application, but if sound synthesis and interaction can be made on par with real objects, it would be strictly more powerful than traditional foley since it could be used to automatically coordinate image and sound, further streamlining the full media production pipeline in analogy with what virtual puppetry is currently doing for character animation.

1.2 Problem domain

In this work we concentrate on the topic of audio-haptic interaction with acoustic models. In particular we will deal with velocity-based interaction, as opposed to position-based interaction more often found in haptics. This section will describe the difference between this and other forms of audio-haptic interaction, problems encountered that are particular to this domain, and finally potential solutions which will be further explored in this dissertation.

Fundamentally this topic is an intersection of two disciplines: haptic rendering, and physically-based sound synthesis. Therefore we will describe each of these separately, with particular focus on the methods chosen for this work, and then discuss how they can be used together.

1.2.1 Haptic and force feedback interaction

Although we often consider the sense of touch as one of our five senses, in fact it is a complex combination of several different kinds of perception: Burdea [18] differentiates between *force feedback*, also called kinesthesia, the sense of opposition to our muscular exertion, and *touch*, referring to the perception of mechanoreceptors in our skin, sensing vibration and pressure. The latter is more often referred to as the *tactile* or *cutaneous* sense [87]. There is also proprioception, the internal sensation of our body's configuration, and vestibular, the sensation of the body's orientation relative to gravity and acceleration [87]. Kinesthesia, proprioception, and tactile perception can together be called *somethesis*, and are differentiated in psychology and physiology from *haptic*, the latter in this case referring specifically to sensing through *active* exploration, though more often in recent literature, *haptic* refers also to passive touch sensation [106].

Even these can be further broken down. Tactile perception comes from several types of mechanoreceptors embedded in the skin which are responsible for different frequency ranges

of vibration [18], and similarly most of the haptic senses really stem from combinations of several types of neural interfaces: for instance, proprioception arguably uses the vestibular system as well as the muscles and skin to help track limb positions, and kinesthesia uses the muscles as well as skin mechanoreceptors to sense pressure [30]. Furthermore, there must be consideration for cross-modal perception, the interaction between all our senses including vision and sound that allow our brain to merge redundant information and form a coherent understanding of the environment [18].

It is clear that a complete description of the haptic sense crosses into areas of biology and psychophysics, and is well beyond the scope of this thesis. Each of these senses requires a different approach for simulation—different signals, different actuators, and different interfaces [87].

Here, we concentrate on a small subset of the haptic sense which is accessible through so-called *force feedback devices*. In this text, we will use the word *haptic* to refer to either active or passive touch sensing through skin mechanoreceptors and kinesthesia. When we use the term *force feedback*, this will be in reference to devices rather than to the human sense of force feedback.

A force feedback device is a robotic device coupled to an operator, usually by grasping, which can sense the user’s input command (position or force), and can additionally exert forces on the user at the point of operation. An *impedance* device exerts forces in response to changes in position, while an *admittance* device is driven to particular positions in response to input forces. By means of a digital feedback loop, it is possible to control the output signal such that it corresponds to how an object would behave, giving the sensation of the object being present. Such an interaction corresponds to a combination of the kinesthetic and tactile senses: kinesthetic, since there is opposition to the operator’s explicit motion, and tactile, since vibration can be exerted through the device’s end effector corresponding to skin-object energy exchange.

1.2.2 Acoustic haptic interaction

This tactile portion of force feedback haptics has been explored to a good extent, but most often in the context of static objects: vibrations resulting from scraping of virtual textures [24, 82], surface friction resulting in stick-slip motion [55], and open-loop modal responses resulting from tapping virtual surfaces [70]. This last was shown to improve the perception

of presence—more convincing surfaces, and enhanced stiffness.

Comparatively little attention has been paid to closed-loop interaction with acoustic objects which react to physical probing by modal vibration. Meanwhile, in reality acoustic vibrations are present in any real-world scenario, and form the basis of many impressions synthesized by simpler models such as the above-mentioned event-based modal responses [70].

Although many problems are shared between acoustic and static tactile interaction, acoustics also presents its own issues since we can no longer think of vibration as an open-loop phenomenon affecting only the operator, but rather it is necessary to consider the energy exchange in the whole frequency range.

On the other hand, higher frequencies do typically have high impedance at the skin boundary, and much of the high-frequency vibration in acoustic phenomena is not perceivable. However, the acoustic reaction of an object may have strong effects in other modalities, especially in the sound they make as air pressure variations are radiated from the surface of the vibrating object. These details thus form the requirements of audio-haptic display of acoustic interaction, which we will examine in more detail in chapter 3.

1.2.3 Audio-haptic interaction

Introducing audio feedback into force feedback simulations is far from a new idea, although the majority of the work on this topic, aside from one particular institution, goes back less than 15 years.

Work at ACROE

Pioneering work in this area must be attributed to researchers at *l'Association pour la Création et la Recherche sur les Outils d'Expression* (ACROE), part of INPG, Grenoble, France. Stemming from philosophical work on the “gesture-sound” relationship between a musician and his instrument, their conception of haptic interaction with physical models, resulting in synchronous output of force, image, and sound, goes back to the late 1970’s [20], and the idea of a *transducteur gestuel rétroactif*, what we call a force feedback device in English, became fully developed in the mid-1980’s [22, 40].

The term “retroactive gesture transducer” is connotative of their conception of the embodied relationship between gesture and object, what Cadoz calls “instrumental in-

teraction” [21]. In this framework, instrumental interaction is fundamental to musical expression, since an object not coupled to a human by gesture is not an instrument. The term *ergotic* was coined by Cadoz to denote the bidirectional energy relationship which is considered a prerequisite for instrumental interaction—the term “instrument” is therefore defined as an object which is involved in an *ergotic* relationship with a human. In this context Cadoz distinguished, as we did in the previous section, between unidirectional interactions with static objects—e.g. texture exploration—and bidirectional interactions involving *manipulation* of objects [19]. The former is an *epistemic* relationship, while the latter is *ergotic*.

The CORDIS-ANIMA formalism was defined, a graphical language used to describe so-called “mass-interaction” physical models [22, 23]. It is a language expressing physical models as collections of mass elements connected by link elements, which are either linear, having damping and stiffness properties, or non-linear, where these properties can be modulated, e.g. by velocity (stick-slip friction) or position (surface contact). This was used as the basis for defining and implementing haptic interactions such as percussion, plucking and bowing [22] materials such as a string, or glass surface [76].

Models were executed on a custom-built signal processor called the Cordis-Temps-Réel, or CTR. Later evolution of this hardware, used in the current work, is described in Appendix A.

Work outside ACROE

Apart from ACROE, during this time and before, haptic technology was developing for more scientific and industrial applications. The 1950’s and 60’s saw development of teleoperation for handling of hazardous materials, necessitated by development of nuclear technology. Typically these devices used cable systems to transmit commands from a “master” device to a “slave” manipulator in another room. When the slave encountered an obstacle, forces would be transmitted back to the master via the same cables.

Eventually, teleoperation saw the replacement of mechanical cable systems with electrical connections, allowing for other types of control such as position-controlled slave servos, thus the slave became a fully functioning feedback-controlled robotic apparatus. Haptic technology for virtual reality naturally arose from the replacement of the slave device and remote environment with a virtual proxy in a virtual environment.

These ideas coalesced into commercial devices such as the Immersion Probe (Immersion Corp. 1993), and the PHANTOM interface [78] developed in the 1990's. These were 3-degree of freedom (3-DOF) serial haptic device for research into force feedback touch through point interaction. Other companies soon followed, and today there are several devices available on the market for force feedback interaction. A good overview of currently available devices, and comparisons of their capabilities and performance based on several metrics can be found in [108].

As a commercial device, the PHANTOM was made to be integrated into commercial computing hardware, and therefore controlled by general-purpose operating systems. Other commercial devices later followed this model. The original PHANTOM used the parallel port, while later devices used Firewire (IEEE1394) or USB busses to communicate with the computer. Each haptic feedback step was computed by a timer interrupt cooperating with the computer's operating system.

The implication is that the timer could not execute faster than the OS scheduler allowed, which typically implied, and still does today, that the maximum sampling rate can be 1000 Hz, since 1 ms is typically the maximum timer frequency available for processes running on general-purpose desktop operating systems. Under simpler, mono-process operating systems like MS-DOS, it was often possible to handle real-time interaction quite efficiently, however multi-process systems such as Unix-likes and later version of Microsoft Windows still have difficulty assuring very precise timing for high-frequency tasks without using special real-time extensions.

Additionally, audio synthesis is generally considered more intensive due to its much higher sample rate, but also more robust to latency, since it is open loop from the computer speakers to the ear. This led researchers introducing audio feedback into haptic simulation to pursue methods different from the purely synchronous approach used at ACROE, where haptic and audio samples are different outputs of the same model, computed in lock-step.

Instead, researchers using PHANTOMs and other devices often executed a feedback model in the haptic timer handler, and used signals and events from the haptic simulation to control a separate sound synthesis model.

For instance, Gillespie [45] created a dynamic-kinematic model of a grand piano action, manipulated via his Touchback Keyboard. While this successfully gave the feel of lifting and releasing the piano hammer, being a piano model the evolution of the string motion was not considered as highly coupled to the manipulandum and therefore synthesis was

performed separately [47].

Other examples include Chu [29], who triggered MIDI messages from events in a haptic environment; Barrass and Adcock [9], who used a granular synthesis approach to simulate the sound of tapping and scraping a recorded object; and van den Doel et al. [129], who used a stochastic model synchronized to a haptic rigid body environment to trigger inter-sample “micro events” in the sound synthesis for virtual Foley purposes during rolling and sliding.

We reproduced some of these ideas related to interacting rigid bodies in a general-purpose run-time programmable environment called DIMPLE [117]. In DIMPLE, one can create simple 3D objects by means of messaging from a sound synthesis programming environment, and react to response messages triggered by their physical interaction with each other and with the force feedback device. In this way, the device is used to push on objects that are fairly decoupled from the sound synthesis, so that the haptic device can control the sound indirectly through interaction.

As for continuous interaction, O’Modhrain [89] and Nichols [85] used haptic devices to rub against a local friction model, and transmitted the velocity and force to parameters of a concurrently executing digital waveguide bowed string model—feedback from the bowed string model to the haptic device was not used, and the two models executed asynchronously at different sample rates. Nichols [85] developed a haptic device roughly in the shape of a violin bow, specifically for this task. This consisted of a cable running through a motorized pulley, such that the user could hold it like a bow and move it back and forth. The final version of this device supported four degrees of freedom to allow up-down pressure adjustments, as well as rotation in various directions for control of bowing parameters and string selection.

Although O’Modhrain and Chafe [90] previously found that the presence of haptic feedback had a positive effect on the playability of a simple Theremin model, studies on the presence of friction on skill transfer for a bowing task did not have clear results [89]. It was suggested that the presence of “noisy” stochastic friction was disturbing to users, as opposed to a “smooth” friction which played a predominantly information-carrying role. However, since the friction did not oscillate in a stick-slip fashion, and was not coupled to the synthesized string motion due to this concurrent asynchronous approach, it may be that the “noisy” friction used in this study was simply not perceptually relevant; although the noise more closely resembled the roughness of bow hair, it added distortion to the

simulation without necessarily carrying meaningful information about the system state, and had little correspondance to what was heard. This conjecture partly motivates our desire in the current work to improve the quality of friction display of acoustic synchronous models, so that we might eventually investigate how various levels of degradation in friction feedback during playing may affect perception and skill transfer.

More similar to the work of ACROE, and also to the work described in this dissertation, was Cymatic, a software program for designing mass-spring systems for audio synthesis [59]. In Cymatic, a force derived from an external source is attached to a mass, thereby allowing control from a human interface device. Conversely, Cymatic allows a “microphone” to be placed on a mass, and used to generate vibration in a haptic device, or to produce audio output. The devices used by Cymatic authors were the Microsoft SideWinder joystick, and the iFeel mouse from Logitech.

This was the only case outside of ACROE that we could find of an environment designed for direct interaction between a haptic device and an acoustic model. However, the choice of technology limited force interaction to lower rates than the audio, thus the lock-step synchronous aspect of the interaction was not present. Nonetheless, in principle Cymatic represented a synchronous acoustic interaction allowing bowing-style interaction with strings, planes, and 3D structures using a force feedback joystick. It could achieve a very similar configuration to that used by Florens [43], although the string was presented as a mass-spring model rather than a modal one; presumably the latter could be achieved, however this is not mentioned in Cymatic articles.

1.3 Bowed string modeling and synthesis

In the above, a couple of string synthesis methods have been mentioned in passing, and in this section we describe them. Although we prefer to think of this work as applicable to any velocity-based interaction with a virtual acoustic object, we use the bowed string as our test case for two reasons: it is a continuation of previous research, which forms a solid basis for comparison between different methods; and it is a clear and well-understood example of continuous-contact musical interaction.

Indeed, with exception to wind instruments, most musical examples of acoustic interaction with continuous contact can essentially be categorized as either bowing, rubbing, or scraping/scratching. We can differentiate these interactions by the following descrip-

tions: scraping or scratching is a tangential motion against a mostly rigid, dry surface, which is usually hard and textured; rubbing is similar but characterised by a stick-slip action, promoted by at least one object consisting of a viscoelastic material (such as skin or rubber), not necessarily textured, and additionally some amount of moisture may also contribute to viscosity at the interface; bowing also includes stick-slip friction, specifically driven by a tensioned set of bow hairs usually consisting of horse hair, and is promoted by a sticky substance on the bow hair called *rosin*. In both bowing and rubbing, if the object has strong natural resonances, such as a string, a metallic bar, or a crystal wine glass, it will respond with a gross oscillating behaviour that dominates the interaction and leads to harmonic motion as described by Helmholtz [58].

As with most acoustic systems, the bowed string can be decomposed into a non-linear *excitation* and a linear *resonator* [125]. The excitation mechanism is the stick-slip interaction between the bow hair and the string contact area. The resonator is the string, which transmits vibrations through the bridge to the body of the instrument. These two portions of the model can be discussed separately.

1.3.1 String model

String motion and the bowed string are among the most well-studied phenomena in musical acoustics [135]. As a non-linear, time-varying interaction which is difficult to examine analytically, numerical modeling has long been an important tool for understanding string behaviour and the nature of bow-string friction [96].

Ideal string motion has been fairly well-understood since Helmholtz [58] described the behaviour of “corners” circulating in two opposite directions as the result of plucking. However, the properties of real strings which exhibit damping, stiffness, and other mechanical characteristics, as well as the complex effects of hair-string contact, have left researchers with a wealth of topics to explore. Woodhouse [133] gives a particularly good summary of various aspects of bowed string modeling.

For this focus on physical accuracy, time-domain numerical methods such as explicit finite difference schemes are fundamental to understanding the time evolution of system state variables. Time-consuming calculations such as the resolution of delay-free loops and application of convolution of boundary impulse responses are needed at each time step [80]. This approach has been used by researchers during the last three decades as a tool

to confirm theory against observation, to help isolate the impact of known parameters, to observe the effects of new problem aspects [96], to indirectly observe internal or difficult-to-measure variables, and to investigate the impact of new knowledge [134].

However, the use of computers for music interaction has also led to a class of simulation techniques that focus on real-time sound synthesis. They emphasize computational efficiency over accuracy, and focus on perceptual aspects of the simulation rather than numerical behaviour—although these goals may often align. These real-time approaches, the focus of this dissertation, can be broken into two main categories: frequency- and time-domain modeling.

The frequency-domain approach characterises materials by their modal response, and a filter-bank (often implemented as an inverse Fast-Fourier Transform, or FFT^{-1} , for efficiency) can be used to calculate the evolution of model harmonics in response to excitation. This has been used e.g. in Modalys [39] simulation software, which can be used to construct a modal model and then excite it using a real-time audio stream.

In the time-domain approach, principally based on the traveling wave solution of the wave equation published by d’Alembert in 1747 [124], energy propagation through a homogeneous medium is represented by “waveguides,” implemented as delay lines combined with filters to represent frequency-dependent dissipation within the medium. Waveguides are coupled by “scattering junction” structures that model how the energy is redistributed (reflected and transmitted) when impedance boundaries are reached. Smith [123] described how to use the digital waveguide approach in real-time modeling of the bowed string. A complete description of the digital waveguide approach can be found in [125].

A third technique is the particle-based approach, as used by e.g. the CORDIS-ANIMA formalism, mentioned above, with a notable implementation being the GENESIS software, although this is intended for off-line simulation [25]. This formalism can be used to represent either frequency- or time-domain approaches. It can be considered a time-domain approach since the state space of coupled particles is integrated through time, their inter-linked configuration representing the dynamics of the material. However, Florens et al. [41] showed that it is also possible to represent a modal model using particles by tuning a set of masses to each desired frequency and coupling them via spring-dampers to a summing “bridge” particle.

An advantage of the particle approach is that incorporating non-linearities, such as temporary links or variable damping and stiffness, can be simple, but a disadvantage is

that since it is a lumped model it may take a large number of particles to achieve good accuracy, decreasing its usefulness for real-time applications; for linear materials the modal variant can help, however, since it shifts the computational trade-off from a spatial particle distribution to a given number of resonant modes. The natural modularity in the uniform representation allows arbitrary mixing of time- and frequency-domain models.

More complete overviews of physical modeling sound synthesis techniques are available from several sources, e.g. [34, 125, 128].

1.3.2 Efficiency and trade-offs

The efficiency of the digital waveguide approach is derived from the separability of propagation and loss enabled by assumptions of linearity in the homogeneous medium. This separability allows the accumulation of continuous losses into a single filter on the output of a delay line, which is responsible only for propagation. Since a delay line is particularly efficient for 1-D structures, due to the ability to implement it as a circular buffer, this technique is excellent for real-time applications where 1-D models can be applied, of which the transverse waves of a string are one example [124].

Computational requirements for the modal approach scale with the number of harmonics represented. On the other hand, modal models can readily take advantage of parallel processing, whereas the time-domain approach is functionally sequential. Additionally they can be made even more efficient by the use of algorithms such as the FFT⁻¹ [44]. Nonetheless, the digital waveguide gives “more bang for the buck” because it represents a time-domain sampling of a traveling wave solution of the wave equation, thus it naturally synthesizes the frequency response up to Nyquist without incurring any per-partial computation [121].

One possible drawback to the use of delay lines is that, in modern computer architectures, memory access patterns are often considered the more important bottleneck in comparison with CPU speed. The use of delay lines incurs memory requirements that scale linearly with the lowest desired frequency, but in absolute terms it is not a lot: a 50 Hz string would require roughly 1000 samples at 48 kHz, easily fitting in a standard 64K level-1 data cache often found on modern CPUs. The circular buffer implementation requires only a single read and write per step, meaning a maximum of two possible cache line misses, where usually the read and write are next to each other. The access pattern is generally

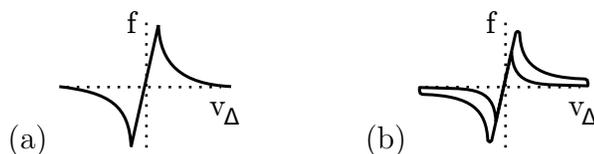


Fig. 1.1 (a) The “standard” hyperbolic friction curve vs. (b) a hysteretical relation. A force-velocity relation such as (b), in which a different return path is followed during stick-slip motion, can be found in experimental observations of rosin tribology, e.g. Schumacher et al. [111]

sequential, which is ideal, although in some instances delay lines may be center-tapped.

In this dissertation we concentrate on the digital waveguide approach, where the exact model is described in detail in section 1.3.4. This selection was made because we were interested in observing the behaviour of the model at high frequencies, and we wished to compare this method with the existing particle-based modal model previously used for audio-haptic interaction [43]. All software implementation used herein is based on the source code of the *Synthesis Toolkit in C++*, a free and open-source software package which provides a collection of real-time instrument models [35]. We shall refer to this software simply as the *Synthesis Toolkit*, or STK. Specifically, we made use of STK’s *Bowed* class, which implements a digital waveguide representation of the bowed string from [123], modified or rewritten as necessary to work on our hardware platform and to implement changes as described throughout this dissertation.

1.3.3 Frictive excitation mechanism

Aside from the specific string model, the mechanism used to excite the resonator is equally important.

The stick-slip phenomenon, which we mentioned is physically promoted by rosin applied to the bow hair, can be characterised by a non-linear function between relative velocity and friction force. In other words, depending on the difference in velocity between the bow and the string, the coupling at the contact surface changes, and this leads to alternating states of sticking and slipping.

We shall refer to this simple velocity-force relationship as the “standard” bowing friction model, depicted in Fig. 1.1a. Due to its shape it is sometimes called the hyperbolic friction curve, since it seems to asymptotically approach zero after exceeding the “capture velocity,” the sharp peak and trough on either side of zero. In measurements of rosin friction it

was observed that a certain hysteresis is in fact present in this relationship [133], seen in Fig. 1.1b, indicating a time-varying viscosity, and it was shown that this was due to a temperature sensitivity of the rosin substance, which in fact melts and refreezes during a stick-slip cycle [122].

Although this hysteresis likely is necessary to achieve certain realistic effects like the “flattening effect” which produces a lower tone at certain combinations of pressure and velocity [134], exactly replicating the friction mechanism is not the focal point of this dissertation. We shall therefore use the standard friction model as found in the digital waveguide implementation in the *Synthesis Toolkit*, however it should be understood that a more accurate friction model could be applied to this work. The standard curve is also used by Florens [43]. Therefore we felt it was prudent to continue to use it, leaving more advanced models of rosin friction for future study.

Serafin et al. [115] showed that a deformable friction model based on rosin temperature tracking improves playability of a real-time bowed string synthesis compared to velocity-dependent models. Therefore future study of this temperature-dependent model may be well-motivated.

1.3.4 The digital waveguide bowed string

As mentioned, Smith [123] proposed a method to model the bowed string using the digital waveguide approach. Here we give a summary of this technique, which will serve as a basis in subsequent chapters for extensions and modifications for haptic interaction.

The right- and left-going directions of travel of the Helmholtz corners in the string are represented as separable traveling waves, v_r and v_l , and are implemented as digital delay lines. We use the variable v to represent the traveling waves since throughout this work we will consider only velocity waves, however, the digital waveguide can in principle be equally used with position, force, or acceleration waves [125].

A block diagram, from [123], can be found in Fig. 1.2. On an instrument, the string is fastened on one end by a *nut*, considered as a perfect reflector represented by a coefficient of -1, while the other end is tightened against the *bridge*, which transmits some portion of the wave energy to the instrument body. The bridge is considered a high-impedance junction, thus this transmission is considered unidirectional and frequency-dependent. Therefore the losses at the bridge are lumped together with propagation loss as the *bridge filter*, labeled

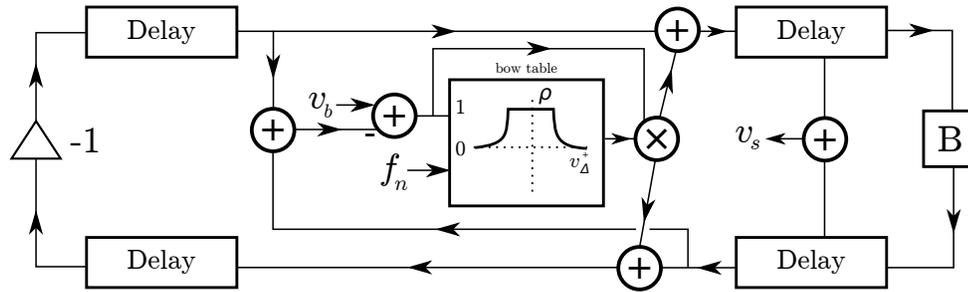


Fig. 1.2 Block diagram of the digital waveguide bowed string, adapted with permission from [123].

B in the diagram.

Both traveling wave delays are split in two at the point where the bow touches the string. This location is usually referred to by a ratio β of the length of the string. The β -ratio has a strong acoustic effect; certain low values of β (bowing closer to the bridge) may produce subharmonics, or produce intra-period spikes due to slipping of some hairs [79]. In practice the parts on either side of the bow can be fused across the terminations, and the reflection functions shifted around to the bow junction, so that only two delays need to be simulated.

The junction itself consists of a transmission coefficient ρ which multiplies the string velocity, determining the propagation or reflection of wave energy depending on whether the bow is in contact with the string. During “stick” periods, $\rho = 1$, forming a perfect reflection. During “slip” periods, $\rho < 1$, indicating a dynamic friction regime, and some amount of wave energy is transmitted to the other half of the string, while the rest is reflected.

The transmission coefficient is actually a function $\rho(v_{\Delta}^+)$ of the difference between the bow velocity and incoming string velocity, and has a shape derived from the standard friction curve.

$$v_{\Delta}^+ = v_b - v_s^+ \quad (1.1)$$

$$v_s^+ = v_{s,l}^+ + v_{s,r}^+, \quad (1.2)$$

where v_s^+ is the sum of the two incoming velocity wave components at the bow, on the left ($v_{s,l}^+$) and right ($v_{s,r}^+$) sides. Following [123], we use the ‘+’ superscript to denote incoming waves at the bow junction, and ‘-’ to denote outgoing waves.

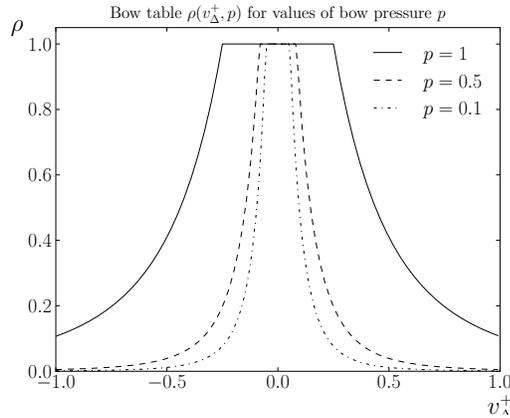


Fig. 1.3 The transmission coefficient $\rho(v_{\Delta}^+)$, for several values of vertical bow force, used to determine wave transmission or reflection at the bow-string junction.

Seen in Fig. 1.3, ρ appears similar to a hat, with a flat portion near zero, and steep slopes that descend toward zero at symmetric positions after the “capture” velocity. Since it is used to calculate the outgoing traveling waves as a reaction to transverse bow force, its shape is derived from the intersection of the normalized bow/string impedance ratio and the load on the string created by the difference between actual string velocity and the incoming traveling wave, for which details can be found in [125]. Neglecting hysteresis, ρ can be pre-calculated as a look-up table or approximated by a simple piecewise function, making for an efficient implementation. Therefore is it sometimes called the “bow table.” The outgoing velocities can thus be calculated,

$$v_{s,r}^- = v_{s,l}^+ + \rho(v_{\Delta}^+) \cdot v_{\Delta}^+ \quad (1.3)$$

$$v_{s,l}^- = v_{s,r}^+ + \rho(v_{\Delta}^+) \cdot v_{\Delta}^+. \quad (1.4)$$

In this work, and in Fig. 1.3, we used the approximation for ρ given in the *Synthesis Toolkit*, which is,

$$\rho(v_{\Delta}^+, f_b) = \min([\lvert v_{\Delta}^+(5 - 4f_b) \rvert + 0.75]^{-4}, 1), \quad (1.5)$$

where f_b is the downward force of the bow on the string.

1.4 Force feedback from the bowed string

Notably for haptics, the above description does not mention the return force of the string on the bow, since there is no need for this during sound synthesis. However, for interaction with a force feedback device, bidirectional communication with the model is necessary.

This was accomplished by Florens [43] using the modal CORDIS-ANIMA model by connecting the device via a bidirectional viscoelastic link to a mass representing the bow. The use of such a link agrees with the theory of virtual coupling proposed by Colgate et al. [33] for enforcing passivity, which we describe in Chapter 2. In the CORDIS model, feedback forces from the string’s modal masses are transmitted back to the bridge, through the bow mass, and finally back to the haptic device.

To use the digital waveguide approach, we must similarly determine the string-on-bow force resulting from the traveling waves at the bow junction. We proposed one method that expressed an elastic connection to a virtual point controlled by ρ , implementing a position-based friction related to the sticking state of the bow-string interface [118]. This model, and an alternative, are described in detail in Chapter 4.

In parallel with our work, Berdahl et al. [13] also published a more general coupling between a force feedback device and a digital waveguide model. The authors proposed gains for elastic and viscous connections between device and model, again agreeing with requirements in teleoperation passivity theory. For plucking interaction, it proposed setting the viscosity coupling gain to zero, with the elastic stiffness at some non-zero value that ensures passivity, and for bowing, the converse was recommended.

This method and details of our implementation of it are presented in Chapter 5.

1.4.1 Noise problems in acoustic friction display

Initially, before developing the method we proposed in [118], a simple velocity-input interaction had been attempted, and we noticed immediately that without proper filtering, noise from the velocity signal significantly decreased the quality of the model output. This was consistent with recommendations in the literature, i.e. that the viscous portion of a damped virtual wall would suffer without a low-pass filter if it was derived from a backward-difference differentiation [32].

However, at high frequency, significant low-pass filtering was needed due to noise, even if a more advanced velocity estimator was used, and we were concerned that this incurred

unwanted delay in the signal path. Moreover, the use of simple virtual coupling to enforce passivity was likely to reduce the overall fidelity of the simulation, since it achieves passivity precisely by limiting the maximum demanded impedance [33].

1.5 Structure of the thesis

From the above, it seems that the main problem with velocity-based interaction is one of *input*—the velocity signal is simply too noisy. On the one hand, this causes introduction of noise into the acoustic simulation. On the other hand, removing it by means of simple filters or reducing its impact by limited-impedance virtual coupling reduces the fidelity of the simulation.

Therefore, a major aim of this dissertation is to determine appropriate methods to either reduce or avoid the effects of this noise source. In particular we wished foremost to reduce perceived noise without sacrificing impedance or responsiveness. In Chapter 2, we review existing approaches to improve velocity-based haptic interaction, and a discussion of the impact of velocity signal quality on friction display is given in Chapter 3.

As our principle contributions, we recognize that the problem can be approached by two orthogonal avenues:

1. Reduced reliance on measured velocity.
2. Improved velocity measurement.

Solutions of the first variety can be thought of as proposals to change the control system structure such that convincingly similar viscous effects can be achieved without requiring direct use of velocity information. We developed a method of driving the digital waveguide entirely by a position-dependent friction model, obviating the need to calculate differential velocity. This method is presented in detail in Chapter 4.

Solutions of the second variety continue to use the velocity signal as input, but depend on improved signal estimation algorithms as well as enhancements to the sensing hardware to improve the quality of the signal that is used to determine the audio-haptic output. The goal is to reduce noise below perceptual thresholds, without introducing unwanted delay. We show that a better signal-to-noise ratio can be achieved, allowing the improvement of overall output quality and increase the achievable impedance range. Several measurement

and estimation methods are described in Chapter 5, and parameter optimizations are used to compare their potential performance in Chapter 6.

Finally, qualitative results from user studies are given in Chapter 7, some general discussion on audio-haptic interaction is presented in Chapter 8, and finally conclusions and future directions are discussed in Chapter 9.

Chapter 2

Background

In this chapter we describe general issues in force display and how these compare to the requirements of acoustic interaction. Methods previously proposed for improving the rendering and display of velocity-dependent forces will be described, including approaches in actuation, electrical and control domains, followed by discussion in this context of our motivation for pursuing the improvement of velocity sensing for digital control.

2.1 Requirements of haptic display

It is well-known that force-feedback display is demanding from mechanical, electrical, and discrete feedback control points of view. A haptic display device (herein referred to as a *haptic device* or *force-feedback device*), to be considered well-performing, must adhere to strong mechanical characteristics. These include both high mechanical stiffness, in order to efficiently relay force and position to and from the end effector, and low inertia, two properties that are at odds [56].

Sensing, affected by noise, and actuation, affected by friction and inertia among other factors, must be precise and accurate enough to avoid generating perceivable artifacts [127]. Meanwhile high acceleration and strong forces must be available to drive the display of rigid surfaces, but the device must also be able to give the impression of zero-impedance motion.

Lawrence and Chapel [71] proposed that a believably rigid surface display must accomplish a stiffness of 10^4 N/m. This implies not only precise control of strong forces, but exceptionally fast response in reacting to displacement thresholds. Fast response, in turn, implies both high spatial and temporal resolution.

In texture display, the frequency requirements are related to the resolution of the texture desired, assuming some limit on scanning speed. To display a texture of 1 mm at a rate of 10 kHz, a resolution of at least 10 μm is required if the scanning speed is only 0.1 m/s [24]. Greater rates and resolutions are required for smaller textures and higher scanning speeds.

Campion and Hayward [24] also pointed out that for high-frequency interaction the mechanical rigidity of a device cannot be assumed as ideal; frequency response at the end effector must be accounted for. Structural resonances and anti-resonances can be attributed to non-idealities in various links along the chain from signal to power amplifier to actuator to mechanical linkages [70]. These can be identified and compensated [70], but some may be difficult or impossible to correct adequately without introducing instability [24].

This problem of non-ideal high-frequency structural resonance is one factor that could impact interaction with acoustic models, since many models are tuned to frequencies of at least 100 Hz and usually produce an extended range of harmonics or inharmonic partials.

Another problem is specific to velocity-driven acoustic interaction, the subject of the current work: Although percussive interaction can be efficiently displayed using open-loop control for rendering of modal responses [70], many musical interactions such as rubbing, scraping, and bowing are friction-based, i.e. they are velocity-dependent and involve continuously modulated contact forces. Friction display has been well-studied in the haptics literature, however, as will be discussed below in section 2.4, most previous work on friction and texture interaction assumes a static or slow-moving environment model. Acoustic interaction must model a system continuously coupled to the end effector with a state changing at frequencies up to 20 kHz.

As will be argued, because the model velocity is dynamic, some rendering methods and hardware solutions that improve friction interaction with static models by reducing dependence on measured velocity information cannot be applied, or must be adapted. Ultimately, non-static model velocity imposes a requirement that device velocity must be known to good resolution, a proposition made difficult by the usual device configuration featuring only displacement sensors.

Making matters more demanding is that in the current work we are interested in audio-haptic interaction; we intend to display the model output to the human ear as well as to the tactile and kinesthetic perception. This imposes further constraints on the importance of signal quality, since the ear is sensitive to a much wider frequency range than the tactile sense. Compared to the tactile range of approximately 500 Hz [18], loud speakers and

headphones have a range of about 10 Hz to 20 kHz. Therefore noise present in frequencies far above the usual considerations for haptics will be audible, and we can predict that humans will be sensitive to an even lower amplitude of broadband noise than in the haptic-only scenario; not only are our ears very sensitive, but in many cases audio will be intended for amplification.

We note that due to compliance in flexible bodies such as strings and membranes, musical acoustic interaction does involve stiffnesses that are generally lower than the rigid surface display requirements listed above. Also, since acoustic resonators are generally tuned to audible frequencies, their frequency range is inherent rather than a proportion of scanning speed like in texture display, which means that the required frequency range for display is a property of the object rather than of the operator.

Nonetheless, most musical sounds feature harmonics throughout the range of hearing, since excitation of musical instruments is non-linear—e.g. the “snap” of a string slipping from a bow hair occurs over less than a millisecond and produces a waveform that features sharp corners [80]. This non-linear excitation leads to similar high-frequency display issues as for textures. Meanwhile, it has been shown that harmonics can be felt by performers, at least during transients and sometimes throughout the duration of a note, and are likely used as information channels during musical performance [26]. This implies that a correct force response during simulation is not only useful for realism, but may play an important role in skill transfer.

2.2 Passive tool display

A common approach in force feedback is to represent the connection between the device and the tool as a virtual coupling. This allows independence from the complexity of the virtual environment model, while allowing a guarantee of passivity. This is possible due to a property of passive systems that passivity is guaranteed when two or more passive systems are coupled; if the virtual coupling, the virtual tool, and the virtual environment model are proven passive, then their interaction can equally be considered passive, and therefore stable [33].

A passive linear virtual coupling is implemented by choosing stiffness and damping gains for a visco-elastic link between the end effector and virtual tool. The virtual tool also has an inertial impedance which represents the mass of the tool, necessary for stability [16].

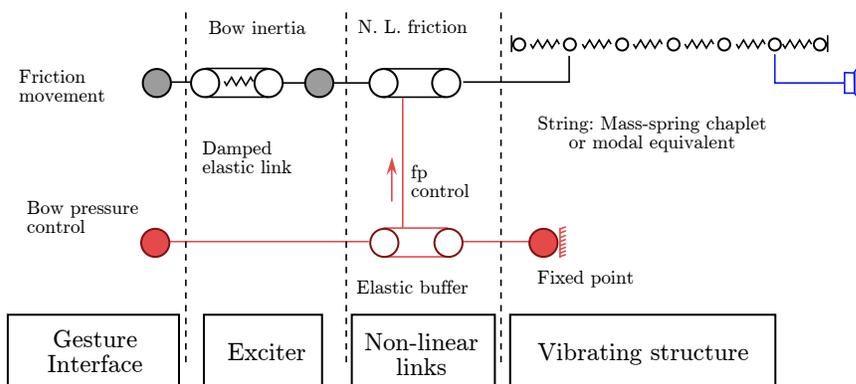


Fig. 2.1 Bowed string force feedback interaction model from Florens [43]. Here a particle-interaction model simulates the string with a series of connected masses. The string is connected via a time-varying damper (“N. L. friction”) to an intermediate mass. This mass is buffered by a viscoelastic spring to the end effector. Bow pressure on the vertical axis, simulated by a grounded spring, also modifies properties of the non-linear link.

A related method is to associate the end effector position with a so-called *god-object*, a virtual point which is forced by hard Lagrangian constraint dynamics to always lie outside or directly on the border of any scene boundaries; that is, at each time step, the god-object is moved instantaneously laterally along the penetrated surface to the surface point forming the shortest distance to the end effector position in the direction of the penetrated surface normal [138]. Thus, when the end effector enters an object, the god-object is stopped, and a visco-elastic link can be used to provide penalty forces.

This has the effect of stopping the “pop-through” effect that can be experienced by simple penalty force calculation when the end effector penetrates past the center of a thin object. Additionally, it has an advantage over pure virtual coupling in that the god-object point follows the end effector exactly in free space, with no apparent inertia, and only becomes dislocated when the end effector passes a constraint boundary. It therefore presents no inertial drag, allowing for fast changes in displayed impedance.

2.2.1 Use in bowed string interaction

In section 1.4, we mentioned two previous works on haptic interaction with bowed string models. Here, we discuss their proposed interfaces between device and instrument models.

The inertial virtual coupling approach was used by Florens [43] for interaction with a

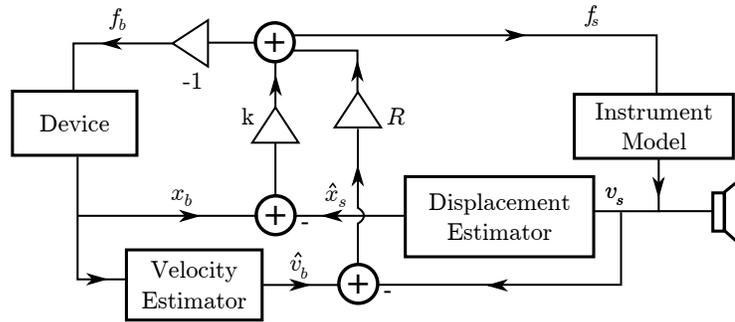


Fig. 2.2 Interaction model proposed by Berdahl et al. [13]. This model, based on teleoperation theory, connects the device position and estimated velocity to an instrument model via spring and damper gains. The model displacement is also estimated to allow position feedback.

bowed string, seen in Fig. 2.1. That model features an intermediate mass between the end effector and the string. The intermediate mass performs two functions: simulation of a virtual inertia, modeling the bow’s inertia, and, as a consequence, low-pass filtering of the device state.

Berdahl et al. [13] also addressed coupling of the end effector with a string model using a spring-damper, this time with a focus on a digital waveguide string model. This proposal for the interface between device and instrument model can be found in Fig. 2.2. In this case no intermediate inertia was proposed. Rather, damping was proposed to be non-zero for friction coupling, with the stiffness set to zero, while the converse was recommended for percussive interaction. In comparison, Florens [43] used non-zero gains for both stiffness and damping during bowing. Prompted by these differences, we investigate the impact of choices for stiffness, damping, and inertia on a virtual coupling in section 3.3 in order to characterise their impact.

In fact, Berdahl et al. [13] discusses several approaches to virtual coupling. Initially, the *direct coupling* approach, in which model force is transmitted to the device actuators and device velocity is transmitted to the model, is rejected, since there is no guarantee that the device can directly deliver the impedance demands of the model. Instead, the paper discusses two main methods for interfacing a device with the waveguide. The first is an *explicit junction*, where the haptic device input/output connections to the digital waveguide are spaced apart by one sample at the bow location, so as to avoid a delay-free loop. This works but, like the direct approach, can give rise to stability issues if impedances do not

match, therefore a virtual coupling is suggested to limit the demanded device impedance. A virtual control point based on estimated position and velocity of the string is coupled to the end effector via a viscoelastic link.

The second is an *implicit junction*, in which the unit delay is removed, and incoming string force is solved for based on known string impedance, taking into account velocity at the junction and force from the damper coupling. Due to the time-varying nature of the damper used to provide the standard rosin stick-slip behaviour, this solution must be recomputed at each step. However, the bow table (ρ) can be used as a solution for efficiency. Since this damper link is bidirectional, an equal and opposite force is also applied to the device.

There is no analogy to the god-object method in either of these approaches, because there is no need for “free space” here. The end effector is continuously coupled to the instrument model. Free movement is nonetheless allowed via dynamic control of the non-linear damping coefficient between bow and string, which is zero when the “bow pressure” is zero—that is, when the end effector is outside the position threshold for string interaction.

An interesting difference is that the Florens model is a completely mechanical approach, and therefore the input signal is passed through an inertia, which provides a filtering effect on high-band differentiation noise. Moreover, when the modal model is limited in the number of resonances, it simply does not respond at high frequencies, further acting as a low-pass filter. In comparison, the Berdahl model relies on an explicit velocity estimator, a categorically more signal processing-oriented approach, which allows for more sophisticated estimation techniques to reduce transmission of noise.

2.3 Influence of signal quality on haptic interaction

In general, signal quality issues, and particularly error due to spatiotemporal quantization, has been discussed in terms of its effect on the stable impedance range, also called the *Z-width*.

Spatial quantization in position tends to be a limiting factor for virtual walls only after considering sampling rate. Colgate and Brown [31] found the following inequality to be a

sufficient condition for passivity:

$$b - \frac{KT}{2} \geq 0, \quad (2.1)$$

where K is the stiffness, T is the sampling period, and b is the physical damping. Therefore decreasing the sampling period T can improve the stable stiffness range of an undamped virtual wall.

This was later expanded by Diolaiti et al. [38] to include the effects of quantized sampling resolution Δ for a device model (eq. 2.2) that included a Coulomb friction¹ component c :

$$m\ddot{x}(t) + b\dot{x}(t) + c \operatorname{sgn}(\dot{x}(t)) = F_H(t) + F_V(t) \quad (2.2)$$

$$\left(b - \frac{KT}{2}\right) + |\dot{x}(0)|^{-1} \left(c - \frac{K\Delta}{2}\right) \geq 0, \quad (2.3)$$

where $F_H(t)$ and $F_V(t)$ are the human input and model forces respectively, and $\dot{x}(0)$ is the initial velocity at entry time into the virtual wall.

This shows that a device with both physical damping and friction can benefit not only from reduction in the sampling period, but also in the spatial resolution Δ . Conversely, there is a limit to the improvements rendered by increasing the sample rate indefinitely if this is not accompanied by greater precision and accuracy.

A similar result was given by Abbott and Okamura [1], showing that the following condition is both necessary and sufficient for passivity of a virtual wall:

$$K \leq \min\left(\frac{2b}{T}, \frac{2c}{\Delta}\right). \quad (2.4)$$

This clearly demonstrates the separability of these two bounds on passivity: device damping requirements are limited by time resolution, while Coulomb friction requirements are limited by position resolution. Not only do both these resolution criteria need to be maximized, but the fact that physical viscosity and friction appear to be needed independently is an interesting result.

The above stability analyses do not include the effects of virtual damping. Colgate and Brown [31] acknowledged that application of virtual damping in the virtual wall model can

¹In the Coulomb friction model, friction force is proportional to the sign of velocity, $F = c \operatorname{sgn} \dot{x}$ [105].

further improve Z -width for an ideal velocity signal. Their full passivity inequality for a virtual wall included a virtual damping coefficient B :

$$b - \frac{KT}{2} \geq B. \quad (2.5)$$

This can be interpreted to show that B improves Z -width, but only insofar as it does not exceed the physical damping of the system. For this reason, they recommended that physical damping be increased, and *negative* virtual damping be used to cancel physical damping during free movement, avoiding this restriction. However, by their acknowledgement this approach can be marginally passive, and difficult to tune perfectly.

In practice, they also acknowledged that backward-difference differentiation introduces considerable noise into the velocity signal, making virtual damping infeasible without further treatment. They proposed the use of a low-pass filter, giving an example of how the introduction of a one-pole filter after differentiation improves the stability margin when virtual damping is in use.

For concreteness, we reproduce their example here. For a damped virtual wall $H(z)$ with a low-pass filter applied to backward-difference operator $\frac{z-1}{Tz}$,

$$H(z) = K + B \frac{z-1}{(\tau+T)z-\tau}. \quad (2.6)$$

From the difference equation for velocity,

$$y(k) = \frac{u(k) - u(k-1) + \tau y(k-1)}{\tau + T}, \quad (2.7)$$

it can be seen that the resolution is,

$$\frac{\Delta}{\tau + T}. \quad (2.8)$$

As the filter parameter τ is increased, the resolution increases, but the filter response time slows. Since introducing delay runs counter to the principles of stability, selection of the filtering operation as a velocity estimator must be considered as a design trade-off between resolution and delay. It follows that an ideal solution would improve resolution while introducing minimal delay.

Colgate and Schenkel [32] continue with a description of the stability inequality for the filtered velocity, which shows a lower restriction on achievable damping as τ is increased:

$$b - \frac{KT}{2} \geq \frac{BT}{2\tau + T}, \text{ for } B \geq 0. \quad (2.9)$$

This shows the limitations of only considering stability, since looking at this inequality alone suggests that τ should be maximized. However, such a choice would ignore that large τ will result in reduced effective damping at high frequencies due to delay. On the other hand, it does also imply that as velocity resolution is increased, virtual damping is more and more effective.

As is evident from the above literature review, most work on the effects of quantization and signal noise concentrate on how it affects the limits of stability; naturally, higher stability limits increase the capacity of the device for presentation of rigid and convincing surfaces. However, a focus in the current work is to point out that passivity and stability do not paint a complete picture of the quality of a haptic display.

For example, it is not the case that a display behaves uniformly throughout the impedance range. Although instability is certainly a phenomenon which can occur abruptly, there is often some amount of abnormal vibration present as the impedance demands are increased. Anyone who has used a virtual wall display of moderate stiffness will also acknowledge that there is a type of noise that can be felt (and heard!) at the position threshold due to the end effector moving in and out of the wall from one sample to the next; this high-frequency noise is distinct from the oscillations introduced by unstable behaviour.

We know of only one work which has addressed this concern. Gillespie and Cutkosky [48] proposed the use of discrete deadbeat control to effectively predict and control when the device enters and exits the wall, such that it can be timed to occur close to the sample time, rather than at a random time between samples. The work acknowledged that position quantization could lead not only to instability, but to perceivable distortion, termed “chatter,” due to this natural timing jitter. While we do not employ deadbeat techniques in the current work, and in general we are not concerned here with virtual wall display, this emphasis on quality of display, rather than stability, remains a focus of this dissertation. We concentrate instead on the quality effects of velocity noise when in control of acoustic models.

2.4 Approaches to velocity-based interaction

This section presents a summary of existing methods for presenting damped and frictive environments using haptic force-feedback technology. Before proposing improvements by the use of velocity sensing, it is first necessary to discuss possible alternatives that allow for stable rendering of viscous effects without requiring high-quality velocity information. The existence of such techniques suggests that we should further motivate our need to improve velocity measurement.

2.4.1 Passive actuator approaches

Previous work has often addressed problems of viscous rendering by alternative means of controlling the *output* forces of the machine, rather than improving the *input* of velocity information.

For example, so-called “passive haptic devices” are able to render forces counter to operator motion by means of controllable physical friction or damping. For controllable friction, electromagnetic elements can be applied to create friction, but often electrorheological (ER) or magnetorheological (MR) fluids are used as controllable rotary brakes.² Originally designed for automotive applications, such brakes have been exploited for display of virtual walls [107], for simulating biological tissue [112], and more recently have been used in a dental implant training system [113]. Two musical uses of MR brakes have been for producing large rebounds of a virtual drum [11], and display of the vibrations of a “scanned” sound file [12].

For controllable viscous damping, Gosline and Hayward [50] proposed the use of eddy current brakes by means of a toroidal magnet for each device axis, which provides an effectively linear braking force in response to input velocity without requiring mechanical contact.

Although these techniques can provide very high-quality rendering of their respective capabilities, they cannot provide a universal solution. For instance, for a damper device to render a Coulomb friction relationship, it is necessary to adjust the damper current in real-time in order to maintain constant friction force relative to user velocity; for this a

²ER and MR fluids are materials, (sometimes actually powders,) that deform under electrical or magnetic stimulation, and can be used to exert lateral pressure on a moving surface, resulting in braking friction.

velocity estimate is necessary. Similarly for a friction brake, rendering of viscous damping also requires real-time following of velocity to control the friction coefficient in order to match a desired linear output relationship.

For dynamic models, neither solution entirely avoids velocity following. Defining a static model, H , as a viscous coupling to the end effector, then end effector force f_h is ideally a function of the device's velocity \dot{x}_h ,

$$f_h = H(\dot{x}_h) = b\dot{x}_h, \quad (2.10)$$

if b is the requested damping. This matches well to the characteristics of the eddy current damper, which we can consider provides the physical damping relationship $f_h = B\dot{x}_h$, where B is the commanded damping coefficient. Using this equivalence between the physical characteristics of the damper and the desired model, we can display it by setting $B = b$. However, for a dynamic model G , f_h becomes also dependent on the model velocity \dot{x}_m , so that,

$$f_h = G(\dot{x}_h, \dot{x}_m). \quad (2.11)$$

Typically, a damping based on relative model/device velocity is required,

$$G(\dot{x}_h, \dot{x}_m) = H(\dot{x}_h - \dot{x}_m) = b(\dot{x}_h - \dot{x}_m). \quad (2.12)$$

Although \dot{x}_m is a known model state, when it is non-zero the equivalence $H = B\dot{x}_h$ no longer holds. Since the above can be rewritten,

$$f_h = b\dot{x}_h - b\dot{x}_m, \quad (2.13)$$

it is possible to render the device-dependent portion of the force using dampers, and make the difference by controlling a motor according to the model state. However, since model force $f_m = -f_h$, updating the state \dot{x}_m requires real-time tracking of the velocity. Without careful design, quality loss in the differentiator will affect motor forces.

Similar issues were discussed by Gosline and Hayward [51] in reference to the rendering of a Maxwell mechanical element, which is defined as a spring and damper in series. In that work, it was suggested that for a selected formulation of the Maxwell element, position x_m

may be updated by an integration operation on device velocity, making it robust to noise. However, they continue, differentiation of x_m , is required to render the model-side viscosity, (the force component of eq. 2.13 rendered by the motors,) risking the reintroduction of noise. This is dismissed by the fact that model viscosity is a short-term contribution during transient initial contact, minimal in comparison with the physical damper contribution. However, for friction-driven acoustics as we are interested in, this is not the case, since the simulated body does not come to rest, and therefore continual differentiation maintains the presence of estimation noise in the force signal.

2.4.2 Electrical approaches

Another proposed method was to use active electrical damping, adding a velocity-dependent analog circuit to the motor command signal by deriving velocity from the motor back-EMF voltage [132]. This approach has the advantage of not requiring extra actuators, and since it is voltage-commanded, has the potential for high-rate control similar to a direct torque command; but it can suffer from instability at high gain due to analog noise, which limits total damping capability. Still, it is superior to a digital solution in-so-far as it does not introduce quantization noise and does not suffer the delays associated with zero-order-hold.

However, in comparison with the flexibility afforded by software, the velocity relationship implemented by such a circuit cannot be modified, and thus imposes the same restrictions as a physical damper when dealing with dynamic models discussed in 2.4.1. Lim et al. [74] proposed that such damping could be frequency-dependent, giving very low damping at gesture frequencies while providing more damping at the high frequencies involved in wall stability. This may be a good trade-off, however the paper expresses concerns that the necessary filtering imposes delay at low frequencies, re-introducing stability problems.

Given a very fast loop rate and high resolution, one could imagine replacing such a circuit with a digital interrupt, allowing for greater algorithmic flexibility; in effect this is the approach we are studying in the current work, however to make the comparison with an analog approach this requires a velocity signal not overly distorted by digitization.

2.4.3 Algorithmic approaches

An alternative method for improving friction rendering has been to formulate algorithms that do not depend on velocity. In particular, Hayward and Armstrong [55] presented a

stateful derivation of the Dahl friction model which expresses friction force as a function of delta position between the end effector and a virtual *anchor point*. This model allows for static pre-sliding, as well as a variety of sliding and stick-slip dynamic regimes. The latter is accomplished by having the anchor point “slip” quickly closer to the end effector after it surpasses a threshold, immediately relaxing the model and preparing for friction force to once again increase gradually.

Although this would seemingly solve the problem of friction rendering, since it allows for high-quality drift-free friction rendering of various regimes, an issue is that this algorithm does not allow for interaction with dynamic models; it assumes a static environment. However, taking a cue from the stick-slip mechanism, it is possible to modulate the position of the virtual point in order to produce effects similar to a dynamic surface. This idea is described in detail in Chapter 4.

This friction rendering method has similarities with the *friction cone* technique described in [53]. The friction cone describes a procedure carried out during so-called “god-object” rendering [138], which keeps the god-point static if it lies within a circular region on the surface, and otherwise moves it directly to the circumference of this region. Since feedback force is generated by a spring between the god object and the end effector position, static friction results from restricting the movement of the point, similar to [55].

The friction cone thus plays the same role as the anchor point in position-based friction, but as it is embedded within the god-object paradigm it is natural to apply friction forces to interacting objects and update the god-object position as necessary based on their movement. In fact, [55] in no way suggests that the same idea cannot be applied to the anchor point, but the god-object method specifies how the point should be updated via Lagrangian constraints in the case of a dynamic environment.

This seemingly works well for a wide variety of virtual environments, but there is one assumption which fails for our scenario, which is that model position is readily available. Acoustic models are one case where, for efficiency reasons, position is not always calculated—and if so, not for the whole spatial extent being simulated. Rather, it is common to simulate propagation of velocity in a medium, which allows for certain important optimizations.

In many cases, position can be integrated at a single point, but tracking position over the whole spatial extent of a vibrating model is often very inefficient. Additionally there is the possibility of position drift due to numerical integration, although methods exist to

mostly avoid it. Use of a leaky integrator is one way to at least ensure that drift does not become excessive [13].

For these reasons, requiring modification of a model to interface with a haptic device can be inconvenient when the coupling is most naturally expressed as a velocity relationship. In particular it violates the modular approach enabled by passive tool display outlined by Colgate and Schenkel [32].

Finally, it should be noted that such a position-based technique, while a good approach to friction display, cannot be used to render linear viscosity, since there is no way to deliver a velocity-proportional force without explicitly determining the velocity. Thus, despite the existence of these techniques, we believe that it is still desirable to understand and improve velocity-based coupling.

2.5 Conclusion

This summary has demonstrated that despite the existence of many techniques to improve velocity-dependent force display, high-quality velocity input is nonetheless a desirable goal. Several choices are available for virtual coupling methods in acoustic interaction, and there are some design choices to be made in the stiffness, damping, and inertial coefficients of the virtual visco-elastic link. The possibility of introducing more complex methods such as estimators and observers for velocity acquisition suggests that a widely-scoped survey of such techniques is needed to establish some measure of what is required and can be expected from velocity measurement.

Chapter 3 therefore discusses the effect of these choices on the perceived quality of the audio-haptic interactive experience, and subsequent chapters explore options for reducing the influence of noise from the velocity signal on the system output, either by reducing sensitivity to velocity signal quality, or by using better estimation and measurement to improve the quality of the measured velocity signal.

Chapter 3

Influence of signal quality on friction display

As discussed in section 2.3 of the previous chapter, most discussions of signal quality in the literature have focussed primarily on stability. Decreases in quality, mainly in regards to the spatiotemporal resolution of position sensors due to digital sampling, can lead to a diminished upper bound on the device's Z -width, the achievable impedance range [31]. This is often expressed in terms of achievable stable virtual stiffness and damping coefficients during virtual wall display.

Passivity plays a large role in stability analysis. Passivity is an energy relationship stating that input energy exceeding storage capacity is dissipated [102, 39]. Since passive systems are stable, and passivity is more easily analysed than stability by tracking energy transfer, it has often been used as a tool for determining stability requirements.

However, it is well-known that passivity can be an overly conservative requirement for stability. This conservatism can stem from unrealistic assumptions about the human impedance, since passivity is true whether the device is free, or a human is holding it with infinite stiffness—two extremes which do not represent the normal use case [2]. Another source of conservatism is an assumption that human operators can apply infinitesimal force impulses leading to instantaneous changes in velocity, which is impossible [1]. Therefore some research has provided arguments for relaxed passivity requirements, allowing higher perceived stiffness. For example, the concept of an “unconditionally stable” virtual coupling can be a less conservative criterion, requiring only that the haptic device be linear time-

invariant, as long as the virtual environment, which may be non-linear and time-varying, is passive [3].

In this chapter, we will argue that stability, while paramount to haptic display, describes only the extreme boundary of displayable impedance, but does not take into account factors in perceived quality of display. Rather, there also exists a threshold for perceivable distortion due to signal error which is in general below the boundaries implied by stability and passivity requirements. Therefore, although passivity is sufficient to prove that a system is stable, it is not sufficient to prove that a simulation is *convincing*; even in an asymptotically stable interaction, users may perceive noise or other distortion which negatively impacts the realism of the virtual reality experience. For audio-haptic simulations which we are concerned with, a high-fidelity sound synthesis output is required, further increasing the need to consider perceived distortion that does not necessarily destabilise the interaction but can nonetheless render the output unsatisfying.

We already mentioned in 2.3 one example of distortion in virtual wall display, known as *chatter*, that occurs at the boundary of a stiff wall. The explanation for chatter is that penetration of the wall does not correspond in a predictable way with the sampling time, and therefore the end effector repeatedly penetrates the wall boundary between samples at an approximately random time, to be corrected with an opposing force at each sample [46]. Similarly the penetration is not detected until a spatial delta is exceeded, further contributing to delay and the uncertainty of the penalty force amplitude.

Under a certain amplitude, chatter does not lead to instability, (in the sense of divergent position and force,) yet it is unquestionably detrimental to realism, since real walls do not *vibrate*. A very stiff wall under the stability threshold may produce significant chatter, felt and heard as broadband noise produced by the device motor, without necessarily introducing any spontaneous bouncing of the handle or uncontrolled motion. A proposal for using a discrete deadbeat controller to significantly reduce chatter was presented by Gillespie [46], which works by using full state feedback to drive the system to a desired position and velocity at wall exit time, making it possible to simulate a predictable continuous-time-like response within the constraints of a sampled system.

In the current work we are concerned with friction interaction between the device end effector and an acoustic model virtual environment. We use the term *acoustic* here to refer to virtual environments that experience deformation and vibration at frequencies equal to or greater than the frequency of operator gestures, as opposed to slow-moving environments

oft-seen in the literature such as clay, soft bodies e.g. used in medical simulation, or rigid bodies, which are typically deformed or displaced at the rate of human gesture. In this chapter, we first give examples of how different types of signal error can introduce perceivable distortion in basic friction display; then, we expand this analysis to include how such distortions are propagated through a virtual tool display to the acoustic model, negatively affecting the quality of the force interaction and audio output. Finally, we provide an outline for some solutions to this problem, to be explored further in subsequent chapters.

3.1 Distortion caused by two types of signal error

Sampled signals contain error comprising many different types, which have different characteristics, and therefore different effects on the haptic interaction and resulting sound properties. These include, but are not limited to,

- Variance,
- Bias,
- Scaling,
- Frequency response,
 - Over/undershoot during transients,
 - Delay,
- Non-linear distortion.

Sources of this error may be any of:

- Analog noise,
 - Environmental interference,
 - Circuit resonance,
 - Non-linear sensors or amplification,
- Mechanical noise,
 - Loose parts,
 - Friction,
 - Mechanical hysteresis,
- Numerical effects,

- Quantization error,
- Numerical error (e.g. filtering, coordinate transformation.)

Here we assume that bias, scaling, and sensor-related non-linearities have been corrected for, and that signal error with the largest effect on interaction may be summarized by two main components: variance and delay. We also assume, as a simplification, that variance around the true value has a normal Gaussian distribution; although not true in general, the main sources of variance will be quantization and analog environmental noise, which tend to be broad in bandwidth. If quantization is selected with high accuracy, so as to produce significantly lower error than a normally distributed environmental analog noise source, then we can assume that a normal distribution is approximated to the degree we require. Exceptions will be noted.

We define the following terminology used throughout the rest of this dissertation:

Noise: the sum of several sources of variance;

Error: any difference between a signal and its true value, i.e. a combination of delay and noise;

Distortion: unintended modifications of output (force or sound) due to error.

In later chapters, “error” will also encompass any other effects potentially introduced by signal processing, such as bias or distortion during transients, however in this section we consider only noise and delay of the sampled signals.

3.1.1 Friction measurements

To study how these two sources of error influence friction display, we needed to characterise the distortion caused by each separately. To this end, a velocity-based friction display using a basic Coulomb friction model was implemented on the ERGOS device. The model used a linear damped portion in the center for stability, shown in Fig. 3.1. Ideally, Coulomb friction, defined as,

$$F(v) = -\text{sgn } v \cdot F_{\max}, \quad (3.1)$$

features an instantaneous jump between the two extrema of force, depending only on direction of velocity, however without this modification, even minuscule velocity noise cause

the model to switch between positive and negative forces rapidly, making the display untenable. As seen in Fig. 3.1(b), a gap was embedded in the viscous region where $F(v) = 0$, so that small noise would not lead to oscillation when velocity is close to zero. We wished to require that users be manipulating the device in order to feel noise-related distortion, so as to ensure that we were examining active feedback rather than passive sensation of vibration.

The slope determined by v_{visc} , v_{gap} and F_{max} should be as steep as possible, so that the maximum friction force is reached quickly for any movement in either direction, but in practise it is limited by resolution, sample rate, and maximum desired impedance. A slope exceeding the device resolution limits leads to limit cycles. In contrast, the size of v_{gap} is determined solely in relation to the noise margin of the velocity signal, as it should be designed such that forces are exerted when the velocity changes, but not when it is near zero.

The region limits of $v_{\text{visc}} = [-0.02, 0.02]$ m/s and $v_{\text{gap}} = [-0.01, 0.01]$ m/s were selected by trial and error, such that no limit cycles were heard or felt up to $F_{\text{max}} = 3$ N, and no vibration was heard or felt when the device was not moving. In practice, since our velocity signal was based on direct sampling of an analog tachometer, described in more detail in Chapter 5, we could achieve a much smaller v_{gap} of approximately 0.001 m/s, however we wished to have some room to increase signal noise without making the display active. The feedback loop was executed at 40 kHz.

Note that in our Coulomb friction model we do not speak of *friction coefficient*, since this term is usually used to refer to a scaling of friction force proportional to the vertical force. Since the device motion was limited to a single axis, this orthogonal force was not available. We therefore speak only of a constant maximum friction force, F_{max} , so that $F(v)$ only depends on velocity. As the force range is expanded, the slope of the viscous relation in the damped region is also exaggerated. The largest F_{max} that can be stably displayed defines the impedance range, or Z -width, of the friction display.

The feedback loop was then modified to add an adjustable level of pseudorandom noise to the sampled velocity signal, and a memory buffer was used to add an adjustable length of sample-aligned delay.

In the following sections, we will first briefly describe qualitative conclusions based on informal observations of these parameters, and then show measurements of the perceptual discrimination thresholds of a small number of individual subjects for these parameters.

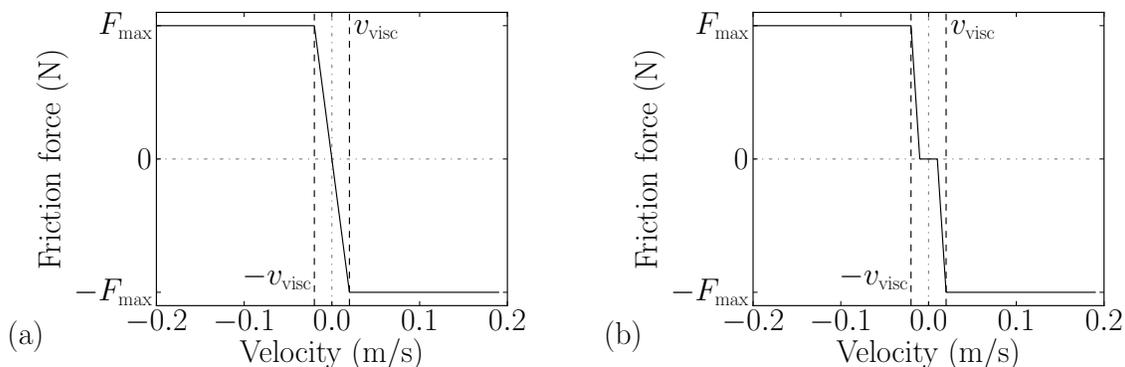


Fig. 3.1 (a) Coulomb friction with a linearly damped section in the center in the range $[-v_{\text{visc}}, v_{\text{visc}}]$, where $v_{\text{visc}} = 0.02$ m/s. (b) Same, but with a gap such that $F(v) = 0$ between $[-v_{\text{gap}}, v_{\text{gap}}]$, where $v_{\text{gap}} = 0.01$ m/s. The latter was used in this pilot study, because it helps avoid vibration when the user is not manipulating the device.

3.1.2 Noise

Noise added to the velocity signal of a friction display produces a sensation that could be described as “raspy,” “chalky,” or “grating.” Additionally it can be heard, since the motors are effectively acting as loudspeakers by moving at random with an amplitude which is physically small, but with enough power to produce audible acoustic air pressure variations.¹

Since forces are dependent on velocity, at nominal levels this noise is only felt and heard when the end effector is moving, see Fig. 3.2. When the user simply rests his hand on the device, he feels and hears nothing.

At higher amplitude, added noise can produce vibration even if the device is not being manipulated. In the case of the gapless damped Coulomb model in Fig. 3.1(a), since forces are proportional to velocity in the v_{visc} region, this occurs when the viscous gain causes force noise to exceed the device’s natural static friction threshold.

Other models are more robust to this effect: e.g. in the Karnopp [63] friction model, static friction is assumed to exist in this small region near zero, rather than exactly at zero—velocity is effectively assumed zero in this region, and any force required to keep

¹Hayward and Astley [56] compared a haptic device to a loudspeaker insofar as it is a well-tuned mechanical system designed for a wide frequency response that should be as flat as possible. It can also, however, be literally a loudspeaker if controlled at high frequency in the audible range, albeit a poor one.

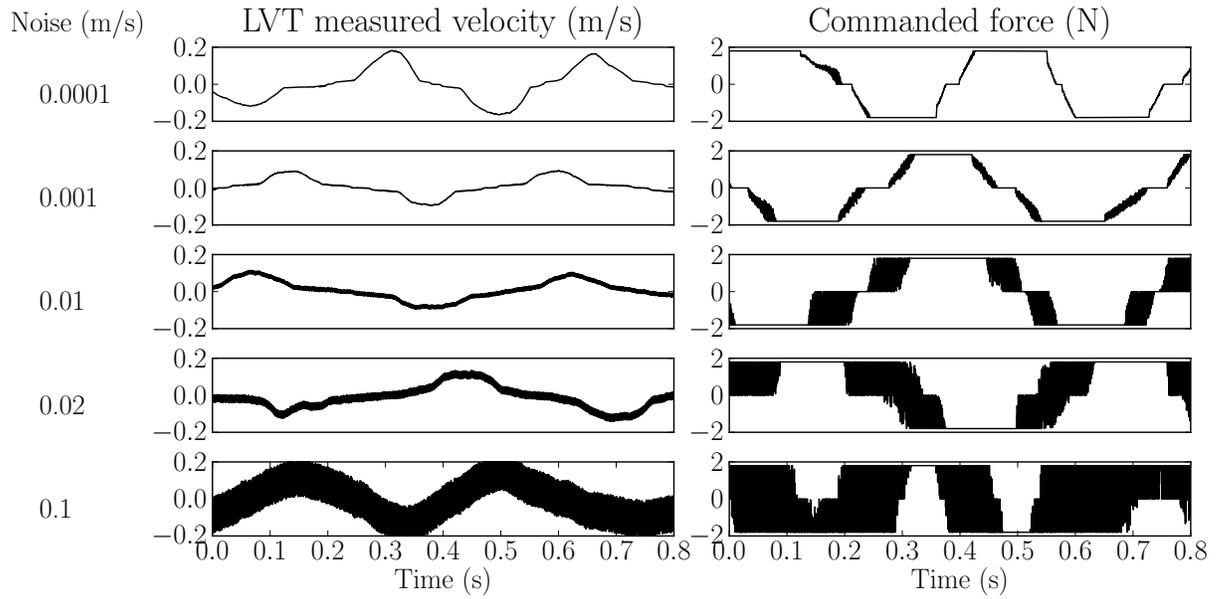


Fig. 3.2 Recordings of velocity with added noise controlling a friction display, i.e. $\dot{x}_{\text{friction}} = \dot{x}_{\text{measured}} + e$, where e is a normal distribution noise signal with zero offset and width from 0.0001 m/s (top) to 0.1 m/s (bottom). Here, $v_{\text{visc}} = 0.02$ m/s, $v_{\text{gap}} = 0.01$ m/s, and $F_{\text{max}} = 1.8$ N. It can be seen that a corresponding level of noise is added to the commanded force during movement, limited by the friction threshold. While velocity is low enough to stay within the gap in the viscous region of the friction curve (e.g. $t = 0.2$ s in top recording, $t = 0.25$ s in second recordings, etc.), or high enough to maximize friction feedback, no noise is produced—if noise amplitude exceeds $\pm v_{\text{gap}}$, the device begins to self-oscillate when at rest, seen in slow periods of the bottom two recordings.

velocity at zero, up to the static friction limit, is assumed to be exerted [104]. This can be rendered in force feedback using a small spring to exert static friction, which also allows for pre-sliding, approaching the Dahl [36] friction model [103]. Since we were interested in examining the detection threshold during active usage, we took a cue from the Karnopp model in designing the gap in the viscous portion of the model to ensure that the device would not vibrate on its own at low velocities. However, we did not render static friction, since we wished to keep the model as simple as possible.

This self-oscillating behaviour does not imply instability—the position does not diverge, leading to out-of-control behaviour. Rather, the device simply rests in place, producing a noise. The effect is limited by the friction curve’s maximum response force, and therefore cannot lead to larger and larger penalties. Nonetheless it is not a pleasant phenomenon and it is desirable to reduce it.

Interestingly, noise can also *decrease* the effective friction applied, since velocity noise exceeding the maximum threshold may produce a lower average force. Notice in Fig. 3.2 that during distorted parts of the middle recording (noise 0.01 m/s) the force noise ranges from maximum to zero, thus the mean force is half of the intended maximum force if the distribution is normal. In the study we describe in section 3.2, we presented subjects with noisy friction, and some subjects had difficulty judging this condition correctly. We were surprised when one subject said that the noisy condition seemed “smoother” to him, which could be partially explained by this effect. Indeed, as one increases noise past a certain threshold, the friction display seems to become easier to move, if one ignores the uncomfortable vibrations being experienced.

3.1.3 Delay

If the velocity signal of a friction display is delayed by a significant factor, the end effector is able to move more quickly than it should, and then velocity-dependent forces appear late and overcompensate, causing a sudden decrease in velocity. This cycle repeats, creating a sensation that could be described as “bumping,” or “chugging.” For smaller delays it might be described as feeling similar to a stick-slip sensation. We noticed that this effect occurs at smaller delays for larger values of the friction coefficient. At very high values of delay and friction, the overcompensation becomes excessive and the system becomes marginally

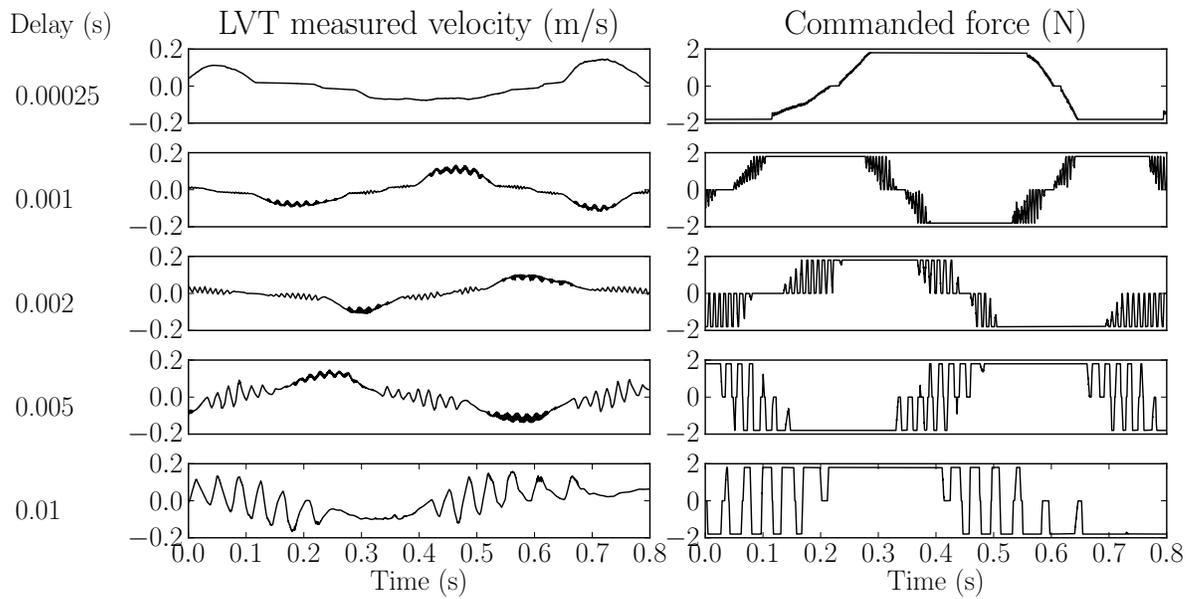


Fig. 3.3 Measured velocity and commanded force for similar gestures interacting with a friction display based on progressively more delayed velocity signal. At low delay, there is little effect, but as delay increases, oscillations begin to appear during movement. At 0.005 s of delay or greater, notice that at low velocities the device never comes to rest—if left untouched, the device self-oscillates, indicating marginal stability.

stable.²

This effect is demonstrated in recordings shown in Fig. 3.3. Progressively more-delayed velocity signals are used for friction interaction, leading to oscillation and finally marginal stability when self-oscillation occurs. In particular we would like to point out that this duty cycle effect is not sudden, but incremental. That is to say, it begins to be apparent to the operator at much lower delays than what we would call unstable, and should therefore be considered a distortion of the friction display, rather than merely a stability concern.

3.2 Experiment #1: noisy and delayed friction display

A study was performed to evaluate the perceptual discrimination thresholds and limits of marginal stability for noise amplitude and delay length in a friction display.³ The same friction conditions as presented in the qualitative description above were used in the study, with $v_{\text{visc}} = 0.02$ m/s and $v_{\text{gap}} = 0.01$ m/s, in order to give room for subjects to increase noise amplitude without creating active vibration.

A log distribution of values for F_{max} were selected, and for each, subjects were given control over the amplitude of either added noise or delay, and asked to identify the minimum values for which,

1. They could detect the stimulus while moving the device;
2. They could achieve sustained vibration.

This allowed us to characterise the region of *distorted stability* for our friction display.

3.2.1 Hypothesis: Velocity-error distortion regimes

Based on the qualitative descriptions from section 3.1, Fig. 3.4 gives a hypothesis for how each error type can lead to different regimes of distortion in the friction display.

²We avoid the use of the term *unstable*, since the constrained nature of friction force does not allow asymptotic divergence of the feedback signal. However, for all intents and purposes, marginal stability, manifesting as large, continual oscillation, makes the system unusable, and we therefore consider it equally undesirable. Whether delay can lead to true, divergent instability in the position signal was not observed, since the marginally stable regime easily causes the boundaries of the end effector workspace to be exceeded—more extreme experiments were not performed for this reason, and for fear of damaging our equipment.

³ This study was performed with the approval of McGill’s Research Ethics Board, REB #105-0908.

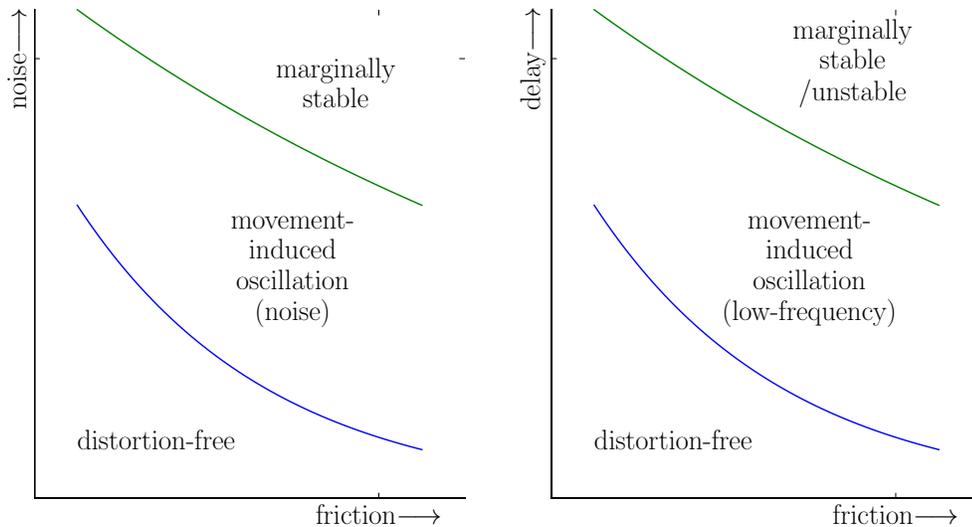


Fig. 3.4 A possible subdivision of different distortion regimes for noise-error and delay-error during friction interaction.

It can be seen that in each case, between “distortion-free” and “marginally stable” there exists a region where we might characterise the display as functional, but perturbed, and therefore unconvincing. Ideally, we would like to determine the parameters necessary to keep the display in this distortion-free region of both graphs, based on an assumption that there exists an area of overlap between them. In the current experiment, however, we will simply show that this area exists and can be identified through psychometric methods. In particular it is informative to notice that increases in both noise and delay lead towards the distorted regime, and therefore simultaneous decreases in delay and noise are necessary to ensure error-free behaviour.

This is significant since many methods to improve velocity estimation, especially low-pass filters, typically reduce noise precisely by imposing delay. The degree to which this trade-off is allowable may depend on the relative sizes of the distortion-free region for each error type, and in turn this may depend on the impedance requirements of the virtual environment, as well as device parameters such as its natural friction and damping, and simulation parameters such as sample rate. For example, it may be different for static friction display as compared to friction interaction with a virtual model; the reason being that the natural behaviour of such a model may mask or enhance the effects of the distortions.

As a starting point, this pilot study characterised the lower bound of this middle region using our chosen equipment with the basic Coulomb friction model that we described in section 3.1.1. The method of finding this bound was to separably determine perceptual discrimination thresholds for noise and delay at several levels of friction force.

3.2.2 Materials and configuration

At the beginning of the study, subjects were seated in front of the TGR force feedback device. They were told to place their right hand on the device handle and their left hand on a MIDI knob.⁴ Subjects manipulated the device programmed with the friction display, and were able to adjust the stimulus amplitude by means of this knob, controlling either the noise amplitude or delay length.

Throughout the experiment, subjects listened to white noise generated by the computer, in order to mask the sound of the device, which was quite audible, particularly for the noise variable. Subjects were provided with a volume knob on an analog audio mixer and asked to find a comfortable level at the beginning of the experiment. They were allowed to adjust this level if it became uncomfortable, however no subjects did so during the experiment. Subjects also wore earplugs, and noise-cancelling headphones were used.⁵

Although this arrangement may not have masked the noisy condition perfectly at high amplitudes, we estimate that the masking noise was adequately higher than the haptic perceptual threshold—all subjects reported that they could not hear the sound of the motors in cases where they could clearly feel it.

Additionally, there was the risk of fatigue, as subjects were repetitively moving the handle against a friction force. Since it was an untimed experiment, subjects were encouraged to take a break between the two trial sets. They were also told they could stop or take a break at any time. The average time was 24 minutes for the first set of trials and 17 minutes for the second set, with a standard deviation of 6 and 4 minutes respectively.

Subjects were graduate students between 26 and 34 years of age, 2 female and 4 male, for a total of 6 subjects. All subjects were right-handed and used their right hands to grasp the device. A photo of the experimental setup being used by a subject can be found in Fig. 3.5.

⁴Akai LPD-8, providing a 7-bit potentiometer.

⁵Bose QuietComfort (QC15).

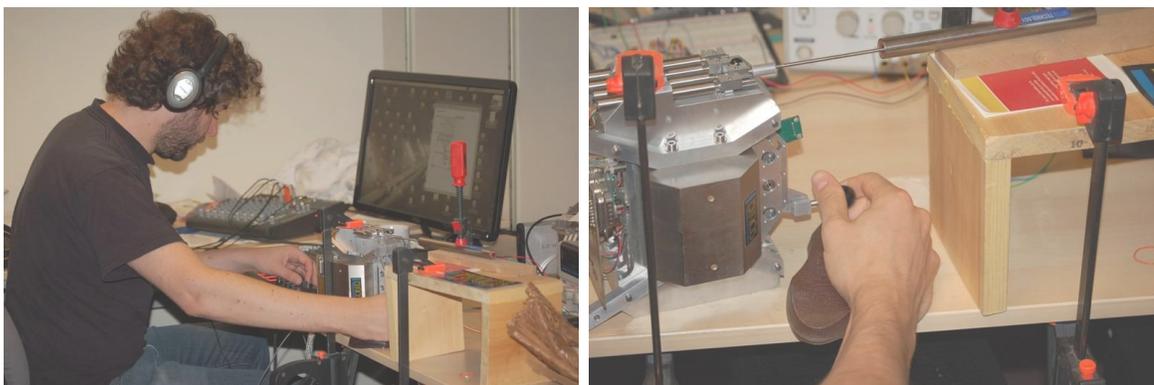


Fig. 3.5 A subject performing Experiment #1.

3.2.3 Methodology

At the beginning of the experiment, a moment was used to discuss the experimental conditions and explain the two stimuli. About 3 to 5 minutes were dedicated to explaining the conditions, and to give subjects a chance to feel the effect of added noise and the effect of increased delay. Subsequently, 4 training trials, two in each condition, were watched by the experimenter to ensure that subjects understood how to identify the stimuli and determine the requested threshold.

30 trials were used for determining the threshold of just-noticeable-difference (JND) of the distortion, (5 friction conditions, 2 stimuli, and 3 samples of each,) after which a short pause was used to explain the marginal stability condition for each stimuli. This was necessary since, although detection of vibration was easy for subjects to understand, in delay conditions the vibration did not always start of its own accord, but rather the limit cycle regime needed to be initiated by moving the device with a sudden small velocity; thereafter the vibration would be sustained without further input from the operator. Therefore subjects were instructed to move the device somewhat during the second set of trials in order to try and stimulate limit cycles, however they were told to ensure that vibrations would sustain while they held the handle still. 4 more training trials were used to ensure that this was well-understood. Subsequently, 30 trials were used for determining the threshold of marginal stability.

Since it could affect the stability margin, we note also that they were told to hold the device relatively loosely, rather than forcing a strong grip, in order to avoid fatigue.

During both sets of 30 trials, the order of conditions was randomized, however a display

on the computer screen was used to indicate whether noise or delay was the current stimulus. This was made available to avoid potential confusion, however after a few trials, subjects generally ceased looking at the screen and used the adjustment knob directly to feel for whichever distortion was presented.

When the threshold was found, the computer keyboard's space bar was pressed to proceed to the next trial. At the beginning of each trial, subjects had to reset the MIDI knob to zero before they had control over the stimulus, to ensure that all trials started in the same condition with no stimulus present. This was also indicated on the screen by displaying the text in red when subjects did not have control, or black when they did have control.

Trials were untimed, and subjects were allowed to explore as much as they wished, but were instructed that no more than about 10 or 15 seconds was expected per condition. They were additionally told that if a decision was too difficult, or, in the second set of trials, if marginal stability could not be discovered, that they were to set the value to zero and proceed to the next trial. Between all subjects, only 4 samples were discarded this way.

3.2.4 Results

Estimated thresholds of detection and marginal stability for both conditions are shown in Fig. 3.6. From this evidence, we can see that the relationship of JND for both noise and delay is approximately log-linear with force gain. Similarly, the margin of stability also follows the same relationship for delay, whereas for noise it is nearly constant, with only a slight downward slope with force gain.

The error bars representing one standard deviation are small compared to the trend, indicating that subjects understood the instructions and that there was little ambiguity in the stimuli.⁶ We conclude that differences between subjects, such as physical sensibility, differing grips on the device, and different choices of response criteria, were less significant

⁶13 of 16 consecutive pairs along the horizontal axis indicated significance by a dependent two-sided student's *t*-test, $p < 0.05$. Those that failed significance were the JND for noise distortion between 1.12 and 1.67 N, and the marginal stability threshold for noise distortion between 1.12 and 1.67 N, and between 1.67 N and 2.5 N. Note however that we are more interested in the significance between the two thresholds than between different friction values, which was significant with $p < 0.05$ for all data. Moreover, lack of significance between friction values for the marginal stability threshold for noise was expected based on the experimental design.

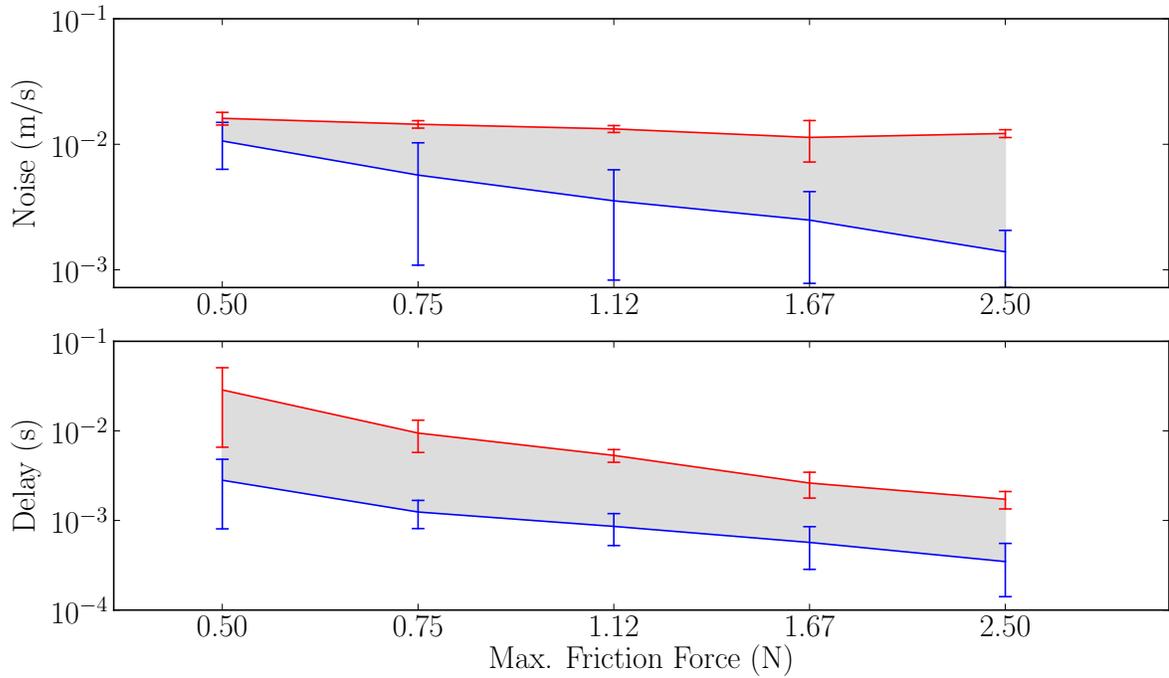


Fig. 3.6 Estimated absolute discrimination thresholds of noise (top) and delay (bottom) added to the velocity signal controlling a friction display, plotted versus maximum friction force. Mean over all trials, 3 per participant for 6 participants, is plotted along with error bars showing one standard deviation. The bottom line is the just-noticeable-difference of distortion, whereas the top line is the lowest point of marginal stability, i.e. where subjects were able to achieve sustained vibration. The shaded region corresponds with the area considered “stable but distorted,” which we note is approximately a full order of magnitude smaller than the limit of marginal stability. For all friction levels in both graphs, the top and bottom thresholds are significantly different according to a dependent, two-sided student’s t -test with $p < 0.05$.

than the distortions of the friction displayed by the device.

3.2.5 Discussion

The shallowness of the noise stability slope was expected, and is due to the fact that vibration begins nearly immediately after the amplitude exceeds v_{gap} , so the threshold is simply a consequence of the choice of $v_{\text{gap}} = 0.01$ m/s for this friction display. Other factors for this threshold are the static friction of the device and the user’s perception of vibration, however in this case these are overwhelmed by the size of the friction gap—nonetheless a small downward trend is evident, showing a small decrease in the threshold with gain.

We selected a large v_{gap} to ensure that users needed to move the device in order to feel vibrations—in other words, we purposely enlarged the region of distortion by choosing a large v_{gap} , so as to help distinguish and visualise it.

In a more realistic setting, v_{gap} would be selected to be approximately equal to the noise margin of the velocity estimate. If $v_{\text{gap}} = 0$, then the device would vibrate as soon as the estimation noise amplitude exceeds the device’s static friction, and the threshold would be principally determined by perceptual discrimination, such that the two thresholds should converge. Here, however, it is made clear that the two thresholds are logically distinct.

The logarithmic trend for the threshold of perceptual discrimination for noise agrees with previous research on tactile perception where amplitude thresholds for detection of sinusoidal stimuli was detected at equally-spaced intervals on a decibel scale [131]. No such research was found on broadband stimuli, although the cited work did show that the presence of a narrow-band noise can increase the detection threshold for a sinusoidal stimuli, known as a “masking” effect. This could be significant for musical scenarios, giving a mechanism by which the presence of noise could decrease the overall fidelity, or subtlety, of the interaction, since larger force amplitude would be required for the operator to distinguish harmonic vibration.

As for the delay thresholds, both detection and stability follow the same log-linear trend with gain. The ratio between detection of distortion and detection of sustained vibration remains approximately constant, with a difference of about one order of magnitude throughout the tested gain range. As opposed to the noise condition, this distortion is not an artifact of specific choices for the friction display, but is due strictly to interaction between input velocity and the viscous gain—it is fundamental to any velocity-dependent

display with significant impedance. It presents a challenge: at our highest $F_{\max} = 2.5$ N, the median is only 0.3 ms, or 12 samples at 40 kHz. At the extreme, some subjects were able to detect distortion at this gain with only 2 or 3 samples of delay!

Since both noise and delay perceptual thresholds decrease with gain, this implies that no particular target for either noise or delay can be selected alone, but must be considered along with the desired display impedance. This has implications for device design, since, if distortion is to be considered, it implies that velocity sensing quality may affect the device's velocity-dependent impedance range, just as position resolution can affect the maximum impedance of virtual wall display.

Conversely, we can conclude that when less impedance is required, requirements may be eased—we can sacrifice some delay in order to filter noise. Requirements are thus discussed in terms of real bowing conditions below in section 3.4.

A brief experience report on psychometric methodology

We chose to use the method of adjustment for this study, since a pilot study showed that inexperienced subjects had trouble differentiating between the stimuli and the device's natural friction characteristics, leading to learning effects and a prohibitive number of trials for the staircase method. Although the method of adjustment depends on self-evaluation, by giving subjects active control over the gain of the stimulus we were able to have confidence that judgements attended to the intended distortion characteristic.

In general this seems to be a good approach for tasks that require active exploration. Because the detection of the stimulus is highly dependent on the user manipulating the device *just so*, without extensive training it can be difficult to have similar confidence that the simulation was properly explored before each judgement is made. This was made evident in our attempt to use the staircase method, where users seemed to change their strategy or their internal criteria once or even several times during the experiment, leading to divergence in the staircase and extending the number of trials. Since the trials themselves were not particularly comfortable, requiring continuous movement against friction and listening to white noise, people who witnessed previous subjects were not encouraged to participate and those who did, did not always finish.

It was evident that this was due to the combination of a complex stimulus—the device having its own particular “feel” in addition to our added distortion—and the uncertainty of

active exploration. Providing active control over the amplitude of stimulus remedied both, and additionally had the advantage of requiring far fewer trials per data point. Our study showed the expected level of statistical significance, allowing us to have confidence in the results.

3.3 Virtual tool display

A “bare,” or “direct” friction display, as used in the above experiment, couples the end effector directly to a friction model by commanding a force related to its measured velocity. As discussed in 2.2, an alternative is the passive tool approach, which introduces an intermediate object that interacts with the displayed environment on behalf of the device. By providing passive coupling between the end effector and this object, passivity can be guaranteed in a modular fashion if the virtual environment is also passive [33].

Thus, friction of any desired impedance can be displayed on the virtual tool by tracking its velocity instead of that of the end effector, and the friction effect is transmitted to the end effector through the coupling, which is stable. We are then left with the choice of how to implement the device-tool coupling, which may for example consist of any form of mass-spring-damper combination that remains passive under the given conditions.

This choice should be made to transmit the desired friction effect, through the tool, at the end effector as transparently as possible. In general, transparency and stability are at odds since the former typically requires high gain feedback.

Beyond stability concerns, we can also consider how these choices may affect the behaviour of the tool in undesirable ways, even though the display is stable: firstly, whether the tool feels “right” according to desired characteristics, e.g. its weight, resonances introduced by the coupling, etc.; secondly, how errors in the measured input signals are transmitted to the tool, and consequently to the virtual environment.

3.3.1 Choice of coupling coefficients

We consider the virtual tool shown in figure 3.7. It consists of three components: an inertia of mass m , a spring with stiffness k , and a damper b . It is connected to a virtual environment through its velocity \dot{x}_t and environment forces f_{env} . Likewise, the device is connected to the user by its position x_d and external forces f_{ext} .

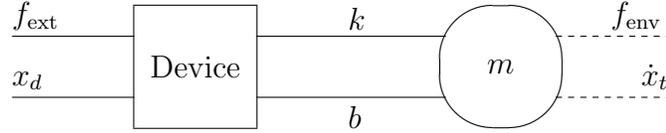


Fig. 3.7 An intermediate mass modeling a haptic tool, which may be connected to an environment via a velocity coupling. Variables are defined in section 3.3.1.

Since we are investigating friction, our problem states that any environmental connection (dashed line) will be velocity-coupled, and possibly non-linear. Therefore the tool system serves two purposes: firstly, as an open-loop system, to transmit the velocity of the end effector to the connected environment; secondly, as a closed-loop system, to simulate the inertia of a real tool. Therefore we shall first discuss how error in the sampled input signals is transmitted through this tool, and then consider how the closed-loop behaviour may affect our choice of coefficients.

Investigation of error transmission

Assuming for now that m is not attached to a larger system on the right, the system response is entirely determined by the properties of the device and the choices of k , b , and m .

The motion of the tool is characterised by the following differential equation:

$$m\ddot{x}_t = -k(x_t - \bar{x}_d) - b(\dot{x}_t - \bar{v}_d), \quad (3.2)$$

where x_t is the tool position, and \bar{x}_d , \bar{v}_d are the sampled position and velocity of the device, consisting of signal plus error:

$$\bar{x}_d = x_d + e_x \qquad \bar{v}_d = \dot{x}_d + e_v. \quad (3.3)$$

This error has a detrimental effect on the minimization of $\tilde{x} = x_t - \bar{x}_d$ and $\tilde{\dot{x}} = \dot{x}_t - \bar{v}_d$, because, while increased gains will normally improve the error between tool and end effector,

large gains also multiply the contribution of signal error:

$$m\ddot{x}_t = -k(x_t - x_d) - b(\dot{x}_t - \dot{x}_d) + ke_x + be_v \quad (3.4)$$

A numerical investigation was performed to determine the optimal values for k and b . Figure 3.8 shows the results of open-loop simulations of the tool system. An initial device state of $x_d = 0$; $\dot{x}_d = 1$ was used, with $m = 1$, and the system was simulated with zero external acceleration. Therefore the tool converged to the device’s constant velocity. A sample rate of 100 Hz was used and the last 5 seconds of a 10-second simulation (1000 samples) was measured for the RMS error of $\dot{x}_t - \dot{x}_d$. 4th-order Runge-Kutta integration was used to closely approximate continuous time behaviour.

In Figure 3.8(a), no error was added to the tool state. It can be seen that in perfectly clean conditions, increasing k and b together effectively provides the best estimate of \dot{x}_d , with best results near critical damping. In Figure 3.8(b), a “measurement error” was introduced by adding a normal distribution of noise in the interval $[-1, 1]$ to the x_d and \dot{x}_d signals before calculating spring and damper contributions to the tool acceleration. This shifts the best-case condition towards $k = 0$, in the slightly overdamped area.

Finally, for an extreme comparison, in figure 3.8(c) the same measurement error was introduced to x_d , but the device velocity for the damper was calculated by backward-difference approximation. As expected, this drastically increases noise in the highly-damped scenario; the scale shows that the lowest error is a magnitude higher than in 3.8(b). However, best-case conditions are still located towards the critically-damped area of the graph with small stiffness.

Note that in both 3.8(b) and 3.8(c), the overdamped areas feature a velocity error with a noise-like distribution, but in the underdamped sections, harmonic oscillation is responsible for the deviation.

We can conclude from this analysis that if velocity is the only output of interest, then a damper-based coupling is desirable in the presence of noise.

Some caveats are that we did not here take into account the perception of harmonic oscillations in the tool coupling, which may very well be imperceptible at some frequencies, nor did we consider the response to disturbances. In particular, high stiffness gains may introduce a phenomenon called “peaking,” a short but very strong overshoot of internal states during initial and transient convergence, which may have detrimental effects [126].

In general this will not happen for pure damper couplings, because estimated velocity is effectly converged using a 1st-order observer. The transfer function from X_d to X_t is,

$$H(s) = \frac{bs + k}{ms^2 + bs + k} \quad (3.5)$$

Therefore, when $k = 0$, the system is reduced by one order.

Display of simulated inertia

The above analysis considers only the open-loop behaviour of the tool. In any implementation, choices of k , b , and m will also affect the closed-loop behaviour, modifying the operator's perception of the tool.

It may be desired to closely simulate free movement, and therefore the effective inertia should be as small as possible. Ultimately it is not possible to make it zero: in practise a minimum mass is required in order to guarantee passivity [16]. However, effective inertia may be minimized within stability limits.

Alternatively, we may wish to modify the device's effective mass, rather than eliminate it, for the purpose of simulation, e.g. holding an object. Brouwer et al. [15] suggested that a modeled inertia is ideally displayed with critical damping, which agrees with the previous numerical analysis. The reason is that any lower choice for a damping coefficient will introduce resonances in the coupling which may cause a sensation of vibration during manipulation. However, this assertion does not make clear what problems may be expected with an *overdamped* coupling. Therefore, here we provide an analytical examination of how stiffness and damping parameters affect the behaviour of a coupled inertia.

For Fig. 3.7, the relationships between forces and positions can be summarized by the relation,

$$\begin{bmatrix} ms^2 + k + bs & -k - bs \\ -k - bs & m_d s^2 + k + bs \end{bmatrix} \begin{bmatrix} X_t(s) \\ X_d(s) \end{bmatrix} = \begin{bmatrix} F_{\text{env}}(s) \\ F_{\text{ext}}(s) \end{bmatrix}. \quad (3.6)$$

Therefore, if we assume $F_{\text{env}}(s) = 0$, the transfer function $G(s)$ from $F_{\text{ext}}(s)$ to $X_d(s)$ is

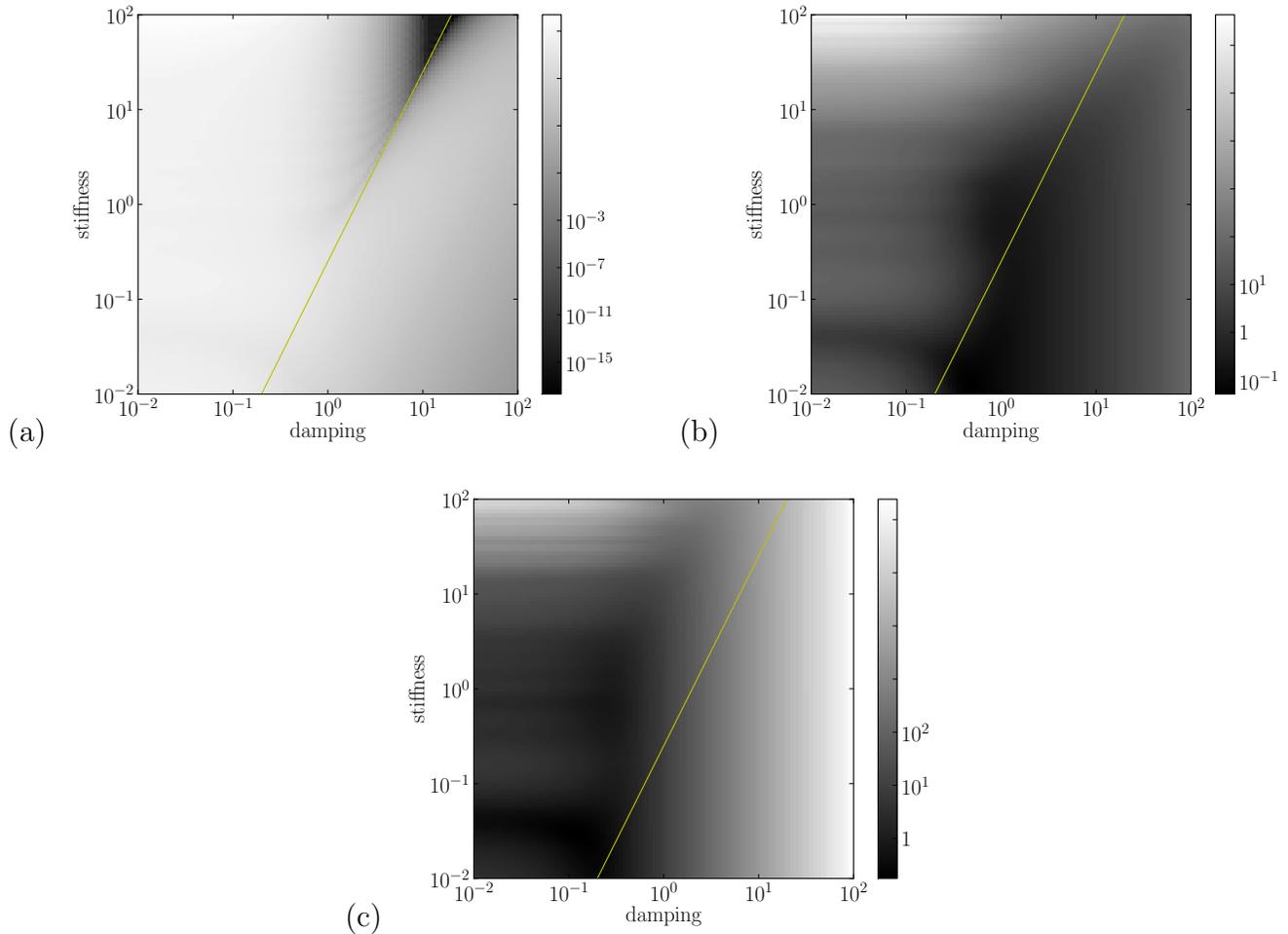


Fig. 3.8 The RMS error of observed velocity for the mass-spring-damper tool model during steady state, as a function of stiffness and damping coefficients. The diagonal line represents the critically damped choice; that is, where $b = 2\sqrt{mk}$. The region to the left of this line is underdamped, while the right is overdamped. (a) For position and velocity with no error. (b) For velocity measurement error of comparable magnitude to position measurement error. (c) For velocity derived by backward-differentiation from the position measurement of similar error magnitude. It can be seen that the lowest-error areas shift from high-stiffness critical damping to a low-stiffness, slightly underdamped system when measurement noise is introduced.

written,

$$G(s) = \frac{ms^2 + bs + k}{m_d ms^4 + (m_d + m)bs^3 + (m_d + m)ks^2} \quad (3.7)$$

$$= \frac{1}{(m_d + m)s^2} \left(\frac{ms^2 + bs + k}{\frac{m_d m}{m_d + m}s^2 + bs + k} \right). \quad (3.8)$$

This shows that the behaviour of the system will emulate an inertia of mass $m_d + m$, with an additional 2nd-order filtering.

Fig. 3.9 shows Bode plots of $G(s)s^2$ under several conditions, demonstrating clearly the mass-related behaviour in the amplitude frequency response of acceleration. It is seen that in low frequencies, the behaviour is of an inertia with mass $m_d + m$, however this behaviour changes to the mass of the end effector m_d as frequency increases. As the coupling strength is increased, a larger frequency band behaves as the simulated inertia.

Comparing figs. 3.9(a) and 3.9(b), it can be seen that the system will happily offset the device inertia by a positive or negative amount, as long as $m_d + m > 0$. The spring-damper link can be adjusted to independently affect the cut-off frequency and resonance; except for extreme values, k and b do not affect the apparent inertia at DC. Note that when $m < 0$, k and b must also be negative:

$$\text{sgn } k = \text{sgn } b = \text{sgn } m \quad (3.9)$$

As a demonstration, we apply the above to the problem of displaying a bow inertia. Our haptic device has a mass of 300 g, but we would like to simulate a 150 g bow. To solve our problem, we could set $m = -0.15$ kg, adjusting the apparent inertia of the device to $0.3 - 0.15 = 0.15$ kg, demonstrated in Fig. 3.9(b). According to the numerical analysis discussed above, we might prefer to use a damper-only coupling. It is seen in Fig. 3.9(c) that such a choice minimally affects the cut-off for high values of b ; it merely removes any resonances.

Finally, as an investigation of nearly total inertia compensation, we set $m = -0.299$ kg in Fig. 3.9(d). (It cannot be set to $m = -0.3$ kg, since the transfer function would be undefined.) Unless k is large, the desired mass is not nearly simulated; however, the required stiffness is accompanied by very sharp resonance. This shows the source of an instability. We do not attempt to find maximum k and b here, since such a determination

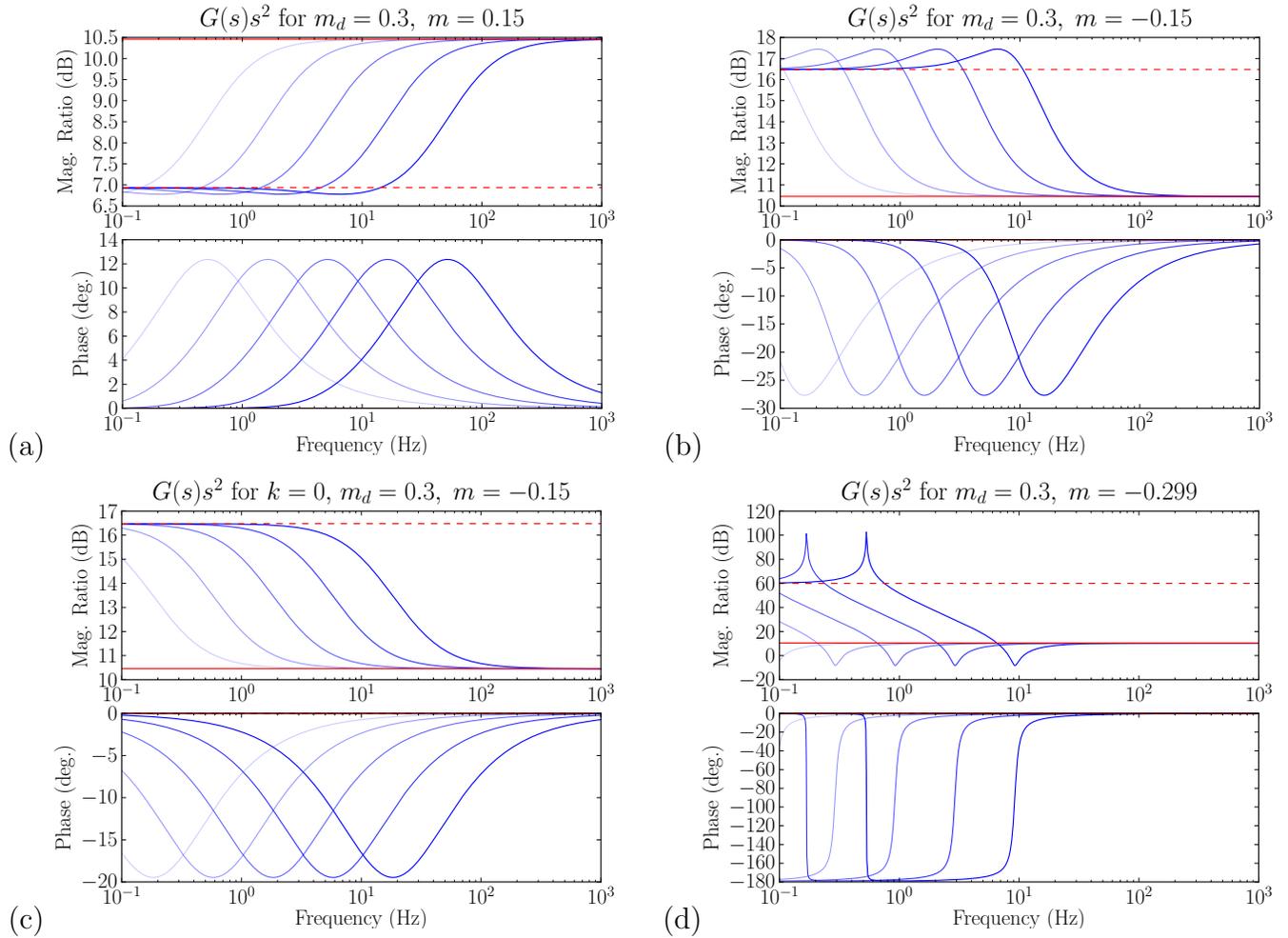


Fig. 3.9 Bode plots of $G(s)s^2$ for the inertial system which modifies the apparent mass of the end effector. We show $G(s)s^2$ here for clarity; the 24 dB/octave drop of $G(s)$ makes visualisation difficult. The solid horizontal line is the response of an inertia m_d , while the dashed line is the response of an inertia $m_d + m$. Values of k range from 10^{-2} (light) to 10^3 (dark). In each graph, $b = 2\sqrt{(m_d + m)k}$. (a) The response of a 0.3 g inertia increased by 0.15 g. (b) The response of a 0.3 g inertia decreased by 0.15 g. (c) The response of a 0.3 g inertia decreased by 0.15 g, $k = 0$, b has same values as in (b). (d) The response of a 0.3 g inertia decreased by 0.299 g.

would be affected by additional factors such as discretization, but we can conclude that the existence of such limits shall dictate a minimum apparent mass.

3.4 Application to bowing—representation of the bow and bow hair

As noted in section 2.2.1, at least two different approaches to device-model coupling have been previously proposed for interfacing with bowed string models, one which includes an explicit tool model as an intermediate inertia [42], and another which coupled the end effector to the instrument through a damped spring [13].

3.4.1 Expected friction force in real bowing

Coupling parameters for the virtual tool are device-dependent since they reflect the impedance ability of the device in terms of maximum force, time resolution, and sampling accuracy. However, the choice can additionally take into consideration the requirements of the virtual environment, in our case being the impedances expected for real bow-string interaction.

Although friction forces at the bowing point are difficult to measure directly, several studies have inferred force signals from indirect measurement, or used specific values in simulation. Several sources in the bowed string measurement and modelling literature were searched to find the maximum parameters for friction force.

While friction force at the bowing point is not always reported directly, we can often surmise from friction coefficient characteristics and reported vertical bow force. It is reported by Pitteroff and Woodhouse [99] that for a violin point-bow model, vertical bow force higher than approximately 0.8 N leads to aperiodic, raucous string motion. For a finite-width bow, this limit value is 1.1 N. While raucous motion is not strictly outside the range of interest for real-time simulation, it does indicate proximity to an upper boundary, which we might consider as some percentage larger, say at 1.5 to 2 N. At this value of vertical force, a maximum static friction coefficient of $\mu_s = 0.8$ was used. Therefore we can consider maximum friction force as about 1.2 to 1.6 N.

Woodhouse et al. [136] measured friction with a rosin-coated, computer-controlled glass rod. They measured forces on the terminations of a violin wirewrap E string and reconstructed bowing-point friction force. Ratio β of position along the string was $\beta = 0.3$,

vertical force was 0.3 N, and velocity was 0.2 m/s. Friction forces up to about 0.1 N were calculated in this manner, with a maximum coefficient around 0.4—it is noted, however, that there is a missing DC component of force, which they estimate at around 0.5 for the coefficient of friction, adding an offset of 0.15 N to the reconstructed force, making approximately 0.25 N for maximum friction force. This makes the maximum friction coefficient about 0.9, for a force of 1.8 N if we consider a large vertical bow force of 2 N as above.

Although 2 N may be a good upper limit to consider for violin playing, of interest also is data specific to particular playing styles. Askenfelt [6] considered bow forces for specific playing styles, choosing a range of 0.5 N for *piano* to 2 N for *forte*. Demoucron et al. [37] similarly found forces of about 1 N for *détaché*, and a maximum of about 2 N for long, accented strokes. A second recording in [37] showed some peaks at almost 3 N during accents. Soft strokes also agreed with [6], with force measured around 0.5 N.

Unfortunately measured force data for other instruments could not be found. Serafin et al. [116] used a vertical bow force of 1.1 N for simulation of a cello D string, with a maximum coefficient around 1. Woodhouse [134] also treated simulation of a cello D string, using a maximum coefficient of 1.2 and found a playable region (non-raucus motion) of vertical force up to about 3 N for low $\beta \leq 0.01$, or about 0.5 N around $\beta = 0.1$.

In general it would be desirable to have more data for other instruments in the bowed string category. For example, we expect the double bass to require higher forces, since the strings are more massive.

3.4.2 The bow hair

In haptic simulation, bow hair dynamics have typically been ignored, however since the hair material represents a compliance at the bow-string interface it is interesting to briefly consider its qualities in relation to virtual coupling.

In numerical simulation, it has usually been considered adequate to model the bow and hair together as a sprung mass [4], and in particular to consider the hair as a massless, clamped spring-damper [97]. This model of a lightly damped, low-mass material for bow hair was justified by experimental observations in [97] of the real and imaginary parts of the complex Young's modulus of the horse hair material by stress-strain testing and vibrational testing. Further, they found good agreement between the spring-damper model and experiment, even when compared with a more realistic viscoelastic continuum model of

the hair material; they conjectured that damping effects due to inter-hair differential motion must have an effect, since higher-than expected damping was required in the continuum model to make it agree with experiment.

This would seem to justify the choice of a spring model for representing the bow *hair*, but what of the inertia of the bow? It is important to distinguish between the mass-spring-damper used as a virtual coupling to the device end effector, which, as we have shown, can be used to model the bow inertia, and the damped spring proposed in physical models for simulating the hair behaviour. If this system is represented as a double-mass virtual tool, it would be structured as in Fig. 3.10(a), which agrees with the model of Pitteroff and Woodhouse [98], Fig. 3.10(b).

However, since bow hair can be considered as massless, we can justify eliminating it. Then, we should consider how to integrate the bow hair spring in terms of the virtual tool. Redrawing the Fig. 3.10(b) without the hair mass, and consolidating the two parallel hair springs into one, since they are merely a visual aid to allow a more literal representation, gives Fig. 3.10(c).

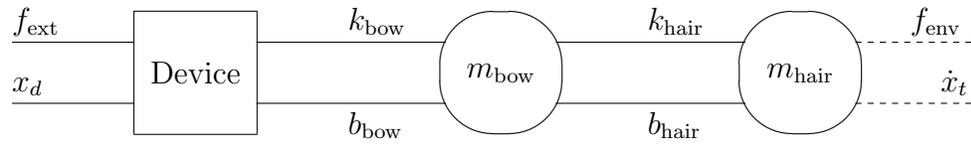
Here, the time-varying rosin viscosity is directly connected to the hair spring. This implies that these elements share a common force, however this is difficult to analyse due to the time-varying, possibly hysteretic nature of the rosin viscosity element. An alternative choice is to commute the hair spring into the virtual tool, shown in Fig. 3.10(d).

We can now demonstrate that if we consider the parameters of the bow hair spring, then the hair model should not have a perceivable effect on interaction. From the device point of view, the hair spring is nearly rigid, and may be considered as part of the bow mass. For example, Pitteroff and Woodhouse [97] settled on values $k_{\text{hair}} = 6 \times 10^4$ N/m and $b_{\text{hair}} = 10$ kg/s (10 N·s/m) for a hair width of 10 mm. This is quite a high coupling strength, reflecting the high tension under which the bow hair is maintained during bowing.

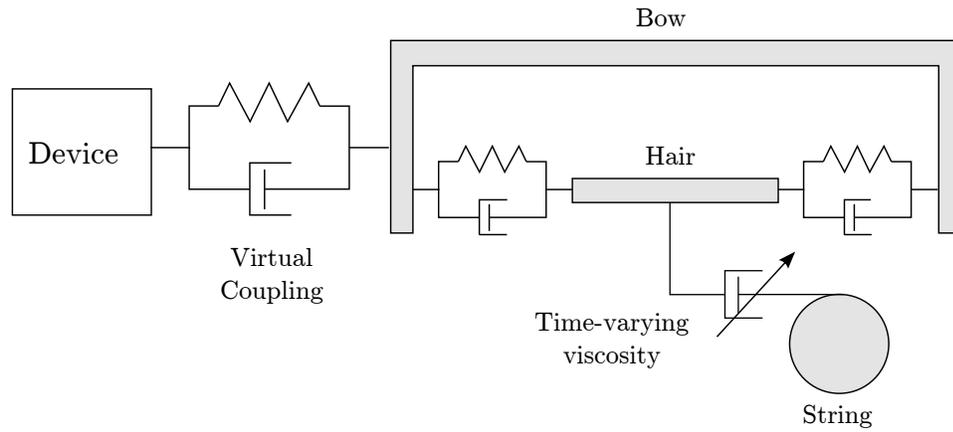
Consider the series spring-damper configuration in Fig. 3.10(d), and for simplicity, let us ignore the damping of the bow hairs, and assume a spring-only virtual coupling with a stiffness of 100 N/m. Then the resulting aggregate equivalent spring would have a stiffness of,

$$k = \frac{k_1 k_2}{k_1 + k_2} = \frac{(100)(6 \times 10^4)}{100 + 6 \times 10^4} = 99.83 \text{ N/m} \quad (3.10)$$

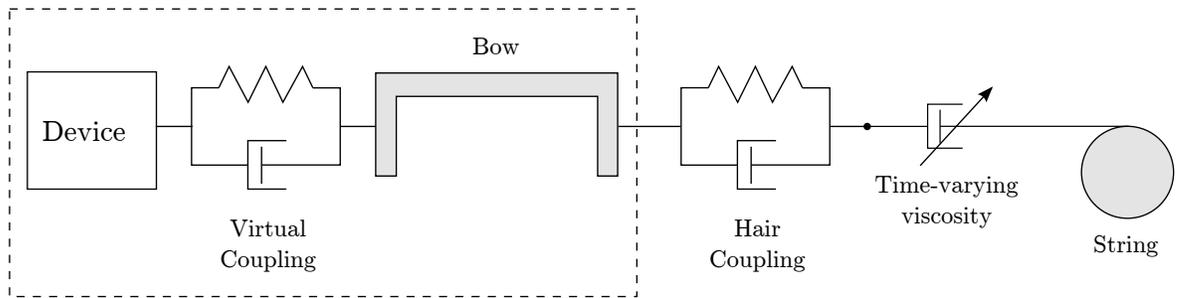
The spring stiffness is changed by less than 1%. For weaker springs the change is even less.



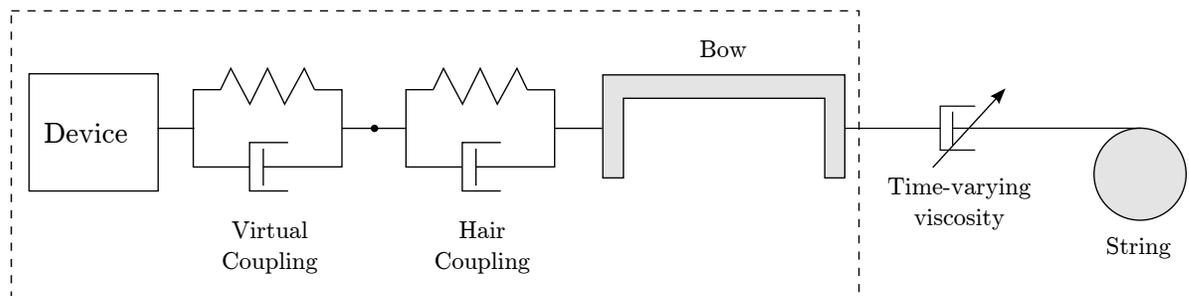
(a)



(b)



(c)



(d)

Fig. 3.10 (a) Two-mass bow and bow hair model. (b) Complete device-bow-hair-string model, adapted from [98]. (c) Eliminating the hair inertia from (b). (d) Eliminating the hair inertia and including the hair spring within the virtual tool.

On the other hand, the hair spring can be considered as part of the acoustic model, since it may have an effect on the high-frequency behaviour of the string. The necessity of including it thus becomes a question for physical simulation but can be essentially ignored from a haptic interaction point of view.

Of course, from a physical perspective, treating the hair as a point interaction is an approximation. It has been shown that the width of the bow can create significant differences in the string velocity waveform under certain conditions, since individual hairs may be under unique stress and may slip at different times [96]. Serafin and Smith [114] showed that a real-time implementation of such a model is possible.

3.4.3 Inertia

As for the bow mass, some details from one bow manufacturer’s web site are given in Table 3.1. Additionally, they state an interesting consideration, perhaps, for multi-DOF simulation:

“The balance of the bow is also an important factor, and can affect the feel of a bow at least as much, if not more so than the weight.” [5]

<i>Instrument</i>	<i>Weight</i>
Violin	58–64 g
Viola	65–75 g
Cello	78–85 g
Bass	125–160 g

Table 3.1 Common bow weights, from Altman Bows [5].

Thus far in the literature, specific modeling of the bow has not been considered carefully in haptic simulation of bowing. Indeed, we do not consider it further in this work, since it begins to form its own set of problems and we chose to focus instead on aspects of signal quality. However, the above citation would suggest that careful consideration of bow simulation in terms of weight and balance may be important to the goal of providing a convincing bowing simulation.

Examining the parameters used by Florens [43]⁷, we calculate 37 g effectively added to the inertia of the device, which is already about 300 g. Therefore we presume this is

⁷Determined via private exchange.

intended mostly for filtering and virtual coupling purposes rather than for simulation of a specific bow inertia. It would seem that bow simulation, including inertia simulation and multi-DOF aspects, is a problem to be address in future work.

3.4.4 Other considerations

Finally, one factor we have not thus far considered is drift. In general a damper-only coupling can suffer from position drift because it is a dissipative element, implying that energy will be lost and velocities will not be perfectly identical. Some amount of spring stiffness can be beneficial for avoiding this, although depending on the environment, for example if the acoustic system is entirely velocity-driven, position drift may not be detrimental. In that case, some small stiffness may still be needed however for numerical reasons, in order to avoid the risk of position exceeding the range of representation.

We note that for the choices used by Florens [43], drift is noticeable at low velocity and moderate pressure, and we informally found it to lead to a loss of realism, despite the presence of a weak spring. One gets the feeling that it is possible to move the bow on the string without exciting stick-slip phenomena, breaking the illusion of sticky rosin. This is a good reason for retaining critical or over-damping with a moderate spring. In this case, however, it may be related to inadequate coupling strength in both stiffness and damping parameters.

3.5 Discussion

In this chapter we have described what we consider to be two important sources of error in sampled velocity signals, and we have looked at how each can distort the display of friction-based interaction and how these effects are propagated through typical virtual coupling methods such that the distortions can be transmitted to the acoustic virtual environment.

In an experimental study, we used a Coulomb friction model to examine how adding noise or adding delay to velocity can create perceivable distortion well below the point where the model becomes unstable or marginally stable. Of course, there exist many other more advanced friction models which more closely model friction in real materials, including effects such as slipping at high velocities, pre-sliding, and elasto-plastic behaviour of static friction. Work by Richard and Cutkosky [104] provides a good summary of friction models applied to haptic rendering.

For this chapter, however, we chose to use a damper-stabilized Coulomb friction, the simplest of friction models in force feedback rendering, exactly because it is very susceptible to velocity noise. Though it may be true that effects described in this chapter may lessen or vary for other models, our goal was not to simulate advanced friction rendering. Rather, we felt that the uncomplicated nature of Coulomb friction would make it easier to draw conclusions based on how subjects perceive distortion. Determining the worst-case results can provide a basis for which to compare the performance of other methods.

We then discussed virtual coupling and showed that although virtual tool display can lessen the effects of instability by enforcing passivity, there is good motivation to use a viscous coupling with this approach, since a critically-damped or overdamped coupling transmits the least error through the tool to the model in the presence of measurement noise. However, this implies that a good velocity signal is needed, otherwise position-dependent couplings must be used, which can lead to unwanted oscillation; we can conclude that virtual coupling is yet another means to trade off noise for other types of distortion. Therefore there remains a need to either reduce the impact of velocity noise, for example by means of more carefully-designed position-dependent strategies, or to reduce the velocity noise itself.

The former approach is discussed in the next chapter, and the latter approach in subsequent chapters.

Chapter 4

Improvement by reduced dependence on velocity

As mentioned in chapter 1, it is possible to derive an expression for a friction model which does not depend on velocity, but is entirely position-based. This leads to smooth sliding friction, free of perceived noisy characteristics, because it does not require differentiation [55]. However, this model assumes contact with a static surface: it can be visualised by imagining an anchor that drags behind the end effector as it is displaced. While the end effector is held still, the anchor is considered static, meaning that for small movements within some radius defined by the expected noise margin of the position measurement, the velocity of the anchor is zero.

We sought to use this friction model to interface with the digital waveguide, hoping to eliminate noise from the velocity differential. In the end we developed two methods, because the first attempt was not entirely satisfactory. However, it nonetheless has some redeeming qualities, and we therefore include a short description of it in this chapter before expanding on what we now consider the preferred method, called `DISTPLUCK`. While the latter provides a cleaner-sounding interaction, it too has some faults, which we leave for the discussion of future work.

We note that the friction model under discussion, derived from Dahl [36] and proposed for haptics in [55], has been previously applied to bowed string modeling, but not for application to force feedback interaction. Serafin et al. [116] explored the effect of this model on the hysteresis of a bowed string model using the digital waveguide approach.

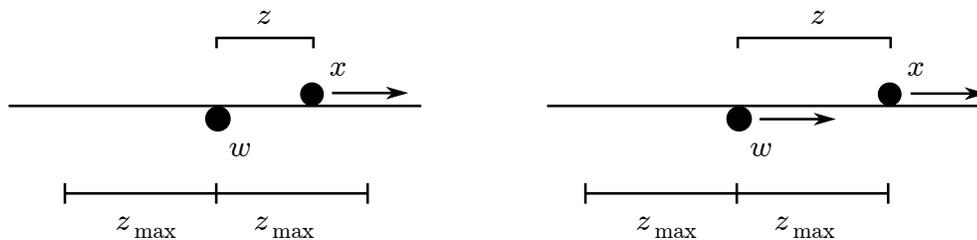


Fig. 4.1 Left: Object position x moving away from adhesion point w and towards the z_{\max} boundary. Right: x has reached the boundary, so w also moves with the same velocity, providing a sliding friction regime.

However, the purpose was to compare its performance to the temperature-based rosin model developed in [134], and was not applied to the problem of robustness to noise.

4.1 Position-constraint friction

Hayward and Armstrong [55] proposed a general friction model, which we refer to as H-A friction, that is robust to velocity noise, and therefore performs well for haptic rendering. It is based on an expression of Dahl friction in terms of displacement instead of a time derivative. Briefly, Dahl's model [36] expresses friction in terms of two points: x , the position of a moving object, and w , the point of adhesion to the surface. The difference between these points, $z = x - w$, represents strain or micro-movement that models pre-sliding during static friction. This small amount of energy is stored due to elastic properties of the materials. When z exceeds a threshold z_{\max} , excess energy is released as the contact point changes, representing a dynamic friction regime, see Fig. 4.1.

In [55] it is observed that in one case of Dahl's model, written,

$$\dot{z} = \dot{x}(1 - \alpha \operatorname{sgn} \dot{x} z) = \dot{x} - \dot{w},$$

the role of velocities \dot{x} , \dot{w} , and \dot{z} are only to express that during sliding, the x and w points move at the same rate, so $\dot{z} = 0$. The same idea can be expressed in terms of position by stating that during sliding, $x = w \pm z_{\max}$ and $|z| = z_{\max}$.

Thus, w is a point established on initial contact ($w_0 = x_0$), and $z = x - w$. As x moves away from w , $|z|$ approaches z_{\max} . When it crosses this boundary, w is adjusted to be on the border of $x \pm z_{\max}$. Establishing a friction force $f = -kz$ provides a smooth sensation

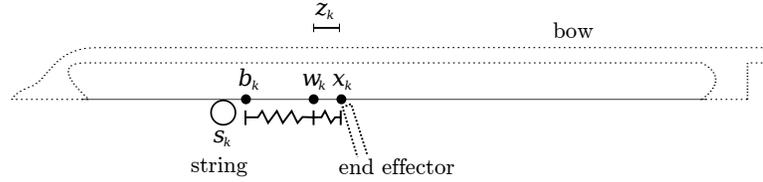


Fig. 4.2 Points used in the TWO-POINT model, representing the end effector position x , the friction anchor w , and the bow point b .

of static and sliding friction. More generally,

$$\Delta w = \alpha(z)z\Delta x,$$

where α is a function specifying friction characteristics. A sliding friction as described can be chosen by setting $\alpha = 1/z_{\max}$ when $|z| > z_{\max}$, and zero otherwise.

4.2 The TWO-POINT model

In our initial attempt to use H-A friction with a digital waveguide string, the anchor point w is connected via an elastic link to a second point b representing the bow position [118]. We refer to this model as the TWO-POINT model.

The bow point is controlled by the stick-slip state of the bow-string friction. During the stick regime, b does not move, since the bow and string are assumed to be mutually adhering, however during slip it is made to follow the end effector position x by a proportion determined by the reflection coefficient. This position difference allows stick-slip friction forces to be transmitted by a virtual coupling between b and x to the end effector during motion. See Fig. 4.2 for a visual description of the relation between these points.

The stick-slip state is determined by examining the waveguide transmission coefficient at the bow junction, called ρ , as described in Chapter 1. Using notation defined in 1.3.4, the value of $\rho(v_{\Delta}^{\dagger}, f_n)$ is 1 during stick, and otherwise it is between 0 and 1, acting as a dynamic friction coefficient; $v_{\Delta}^{\dagger} = \dot{w} - v_s^{\dagger}$ is the difference between the anchor point and the incoming string velocity; and f_n is the normal (vertical) force of the bow on the string, calculated from a virtual wall in the orthogonal axis. We use \dot{w} instead of \dot{x} or \dot{b} to calculate v_{Δ}^{\dagger} , because using \dot{x} would go against our initial goal of avoiding the end effector velocity, and using \dot{b} would create a circular dependency, as the sticking state would depend on b ,

whose motion in turn depends on the sticking state.

Since ρ controls the differential slippage between the bow and string, then $1 - \rho$ can be used as a coefficient to control whether the bow point moves with the end effector, or if it remains in place, presumably dragged by the string:

$$\dot{b} = (x - b)(1 - \rho) \quad (4.1)$$

or in discrete time,

$$b_{k+1} = b_k + (x_k - b_k)(1 - \rho) \quad (4.2)$$

$$= x_k - (x_k - b_k)\rho. \quad (4.3)$$

By this rule, the bow is stuck in place when $\rho = 1$, but otherwise may move with the end effector, simulating slippage. Additionally, ρ can be used to modulate friction forces on the end effector, since it is effectively a friction coefficient:

$$f = \rho f_n(x - b). \quad (4.4)$$

4.2.1 Discussion

This model has some properties that users found interesting. When the end effector is dragged with light-to-moderate normal force, the sensation of stick-slip friction is clear. Pushing harder into the string gives a “sticking” feeling which was not entirely unlike the feeling of pushing too hard with a rosined bow. Finally, pushing *very* hard allowed the bow and string to stick completely, since the system degenerates to a bidirectional spring. Users found this entertaining because with some practice, one could lift the bow up quickly and actually pluck the string, let it ring, and then damp it by placing the end effector back on the string.

However, as a noise removal method, this approach left something to be desired. Although it did effectively remove noise when the end effector was not moving, which was certainly problematic when taking the end effector velocity directly, this model still requires a differentiator for estimating \dot{w} . Since w is a virtual point, the velocity is exactly zero when the end effector is stuck, however during movement, the end effector velocity is transmitted perfectly, along with its noise. Thus, the use of \dot{w} acts more to “gate” the noise than to

actually remove it.

It represents an improvement, since with a good velocity estimator the remaining noise is somewhat masked by the sound of the model, but it is not entirely satisfactory. The next section describes a different method to drive the waveguide based on noise-free impulses driven by the friction model, rather than taking any velocity signal as input.

4.3 The DISTPLUCK model

As mentioned, one assumption in the H-A friction model is that w is a point on a static surface. In this work, we effectively attach w to a dynamic model, a string, causing modulation of friction forces based on distance $z = x - w$. This friction energy is transmitted to the string, to be delayed and reflected, eventually affecting later motion of w , which leads to periodic motion. Since friction forces, determined from the distance between two points, lead to velocity impulses that stimulate the string like a series of plucks, we refer to this model as DISTPLUCK [120].

By starting with a position-driven friction model that has good properties for generating noise-free forces, we can be sure not to transfer noise to the resonator velocity. The difficulty is in the feedback path—determining how to adjust the friction anchor point in a way that is coherent with the resonator model. For instance, care must be taken to limit the anchor movement: if it is moved ahead of the end effector position it will create active forces which can generate energy, potentially leading to instability.

In DISTPLUCK, bow-string friction is synthesized using an H-A friction model as in TWO-POINT, however the anchor position is chosen to be related to a constrained integration of the string velocity. Sliding occurs when a maximum tension is exceeded, indicating the transition from a static to a dynamic friction regime, thus sliding is used to determine the timing of impulsive energy into the string model that results from restoring force acting on the string after release.

We call the anchor point s , representing the string position constrained to bow movement. The end effector is considered as the bow position b . Very small bow movements lead only to pre-sliding, and do not allow the string to slip, but if the distance $z = b - s$ exceeds a small region z_{\max} , then the positions should be adjusted to ensure that $|z| < z_{\max}$. This proximity constraint models the fact that the bow contact position changes during slip.

Either s or b could be displaced to constrain z ; we choose to adjust s since this can be

used to simultaneously track bow slip and string motion, minimizing the number of state variables. When slip occurs, the bow experiences a sliding friction in opposite direction to bow movement. The sliding friction force is related to the normal force component by:

$$f_b = -f_n z, \quad (4.5)$$

where f_n is the normal force component. We obtain f_n by applying a virtual wall model $f_n = kx_n$ when $x_n < 0$, $f_n = 0$ otherwise, on the vertical axis orthogonal to the bowing direction, such that the device displays the penalty force $-f_n$ in the same axis.

Additionally, the string is affected by a restoring force f_r . Corresponding velocity $v_r = (f_r/m)\Delta t$ is added to the string, where m is the linear mass of the string. This force has the same sign as bow friction, opposite to bow motion. When sliding occurs, v_r must describe the release velocity of the string, which, as will be shown, depends on its deflection ΔL , as well as the bowing point along the string and the string tuning.

Note that due to the constraint on x_s , we are not tracking string position at the bowing point, but rather the constraint ensures that z is proportional to bow-string tension. Therefore the full deflection of the string is not known. However, release occurs at maximum tension, when $|z| = z_{\max}$.

We now assume that maximum tension corresponds to a maximum deflection of the string ΔL_{\max} which is proportional to z_{\max} . We define a proportion λ , such that $\Delta L_{\max} = \lambda z_{\max} f_n$. The string's maximum deflection is related by λ to maximum tension and normal force. We can essentially choose λ such that it scales impulses of f_r as desired, since this is similar to choosing the friction coefficient of the bow-string contact.

Restoring force for small string deflections is the tension T times the curvature of the string $\sin(\theta) \approx \tan(\theta) \approx \Delta L/L\beta$, where β represents the position of the bow-string junction as a fractional of the string length. The total restoring force f_r is the sum of both sides,

$$f_r = \frac{\Delta L T}{L\beta} + \frac{\Delta L T}{L(1-\beta)}. \quad (4.6)$$

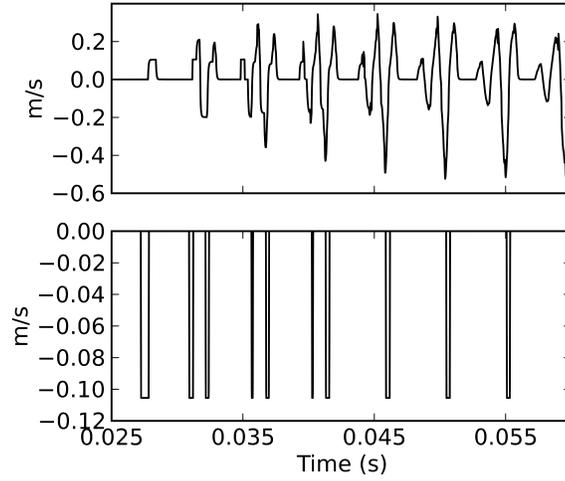


Fig. 4.3 String velocity v_s (top), resonator input signal v_r (bottom). Impulses begin in an irregular fashion, but quickly the string behaviour dominates and leads to periodic timing.

Substituting the string tension $T = mL(2F_0)^2$, and $\Delta L = \lambda z f_n$,

$$f_r = \frac{m(2F_0)^2}{\beta - \beta^2}, \quad (4.7)$$

$$v_r = -\lambda z \frac{(2F_0)^2 \Delta t}{\beta - \beta^2} f_n, \text{ when } |z| \geq z_{\max}. \quad (4.8)$$

The simulation parameters are F_0 , Δt , λ , and β , which can be lumped together as a single coefficient γ , giving,

$$\gamma = 4\lambda F_0^2 \Delta t / (\beta - \beta^2), \quad (4.9)$$

$$v_r = -\gamma z f_n, \text{ when } |z| \geq z_{\max}. \quad (4.10)$$

This added velocity, whose amplitude during impulses will be effectively equal to γ , is propagated in the wave, and eventually pulls the x_s integrator in and out of the z_{\max} region when it reflects, creating stick-slip motion. Fig. 4.3 shows the resonator input signal generated by friction force developing into a series of periodic plucks. The shape of impulses in $v_r(t)$ are square, therefore no measurement noise is transmitted to the resonator; only some aliasing noise may be present.

As with H-A friction, the $|z| \geq z_{\max}$ condition can be more elegantly taken into account

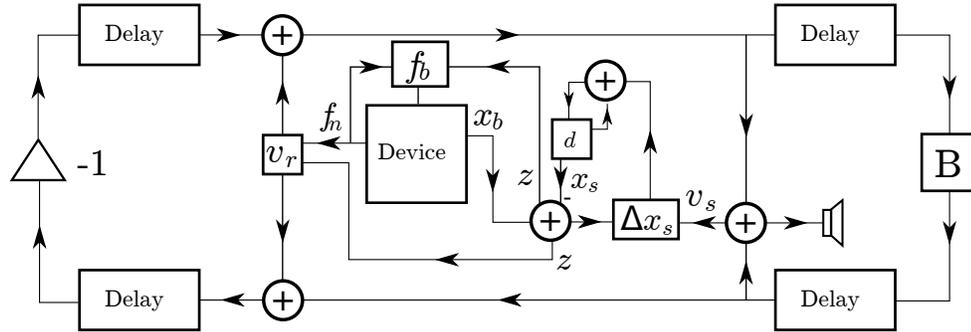


Fig. 4.4 The DISTPLUCK model with digital waveguide resonator. The formulas for Δx_s , v_r and f_b are shown as unexpanded functions here, and are defined in sec. 4.3. The d block prior to x_s is a unit delay. The device provides position b and normal force f_n ; the latter is determined by a simple virtual wall model in the vertical axis, not shown here.

using a function $\alpha(z)$. A difference from H-A friction is that after integrating string motion, $|z|$ can temporarily end up some distance beyond z_{\max} . Therefore instead of integrating \dot{x} as in the H-A model, it is necessary to adjust x_s by a delta distance $z - z_{\max} \text{sgn } z$ whenever $|z| \geq z_{\max}$. This implies the following update to x_s :

$$\Delta x_s = \alpha(z) z_{\max} (z - z_{\max} \text{sgn } z) + v_s \Delta t. \quad (4.11)$$

That is, x_s is moved closer to $b \pm z_{\max}$ by a proportion determined by α , plus the string's change in position.

Finally, v_r is also scaled by $\alpha(z) z_{\max}$ to indicate that the string responded to a ratio of the full restoring force, making the $|z| \geq z_{\max}$ condition implicit:

$$v_r = \alpha(z) z_{\max} \gamma z f_n. \quad (4.12)$$

A block diagram of the described algorithm can be found in Fig. 4.4.

4.3.1 Partial slip provides anti-aliasing

It was mentioned above that the rectangular shape of the pulses during slipping that drive the resonator will lead to aliasing. Here we detail an approach towards accounting for aliasing by taking into account partial slip.

Describing dynamic behaviour in terms of proportions based on $\alpha(z)$ allows consid-

eration of other choices for friction. For instance, [55] considers a *stick-creep-slip-slide* configuration, where $\alpha(z)$ slowly approaches $1/z_{\max}$ instead of being discontinuous:

$$\alpha(z) = \frac{1}{z_{\max}} \frac{z^8}{z_{\text{stick}}^8 + z^8}, \quad (4.13)$$

where z_{stick} is a region just inside z_{\max} defining the boundary between stick and slip. Here, we are not considering the contact presliding stick-slip motion considered in [55], since net displacement is, in the case of bow-string contact, dominated by the string behaviour. However, the *creep* region represents a valuable physical effect because it establishes a dissipative regime where the bow and string may be partially slipping, comparable to the slope on the sides of Smith's reflection coefficient ρ .

For the present purpose of simulating rosin-like friction the slope is not sufficient, creating a small velocity bias that fails to create a stick-slip behavior. The function can be generalized to tighten the slope, dropping z_{stick} :

$$\alpha(z) = \frac{1}{z_{\max}} \frac{z^n}{z_{\max}^n + z^n}, \quad (4.14)$$

taking n to high values such as 40 or more, see Fig. 4.5. This function more closely approaches a discontinuous α , but gives an initial velocity bias that builds up to a series of impulses more gradually, increasing the attack time. The pulses themselves are more rounded, providing significantly more roll-off towards the Nyquist frequency which helps avoid spectral fold-over due to non-band-limited pulses [100], see Fig. 4.6. To allow for high-rate real-time execution, a piece-wise linear approximation of the rounded α is used in both the offline and online simulations, with the curved regions within positive and negative $z_{\max} \pm 0.01$ mm composed of 100 linear segments, and the rest set to 0 and $1/z_{\max}$ respectively.

4.3.2 Offline simulations

The *Synthesis Toolkit's* BOWED class, which is a straight-forward implementation of a digital waveguide model, was compared to the TWO-POINT, and DISTPLUCK models using offline simulations at 40 kHz under noise-free and noisy conditions. The input position signal is intended to mimic a bow stroke starting at zero and reaching a constant velocity of 0.02 m/s (Fig. 4.7d). In the noisy condition, simulated noise is added to the position

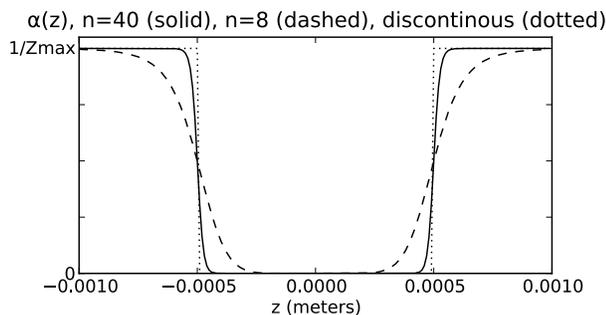


Fig. 4.5 Curves denoted by different versions of $\alpha(z)$.

signal with an amplitude of $0.6 \mu\text{m}$. Velocity is derived by backward difference, leading to a velocity noise amplitude of 0.024 m/s .

Using a real haptic device, of course, the input would be affected by the displayed bow friction force. We have opted to simulate in open loop since this work is primarily affected by input signal quality: since the control signal is slow-moving, inserting a second-order device model would only low-pass filter the noise.

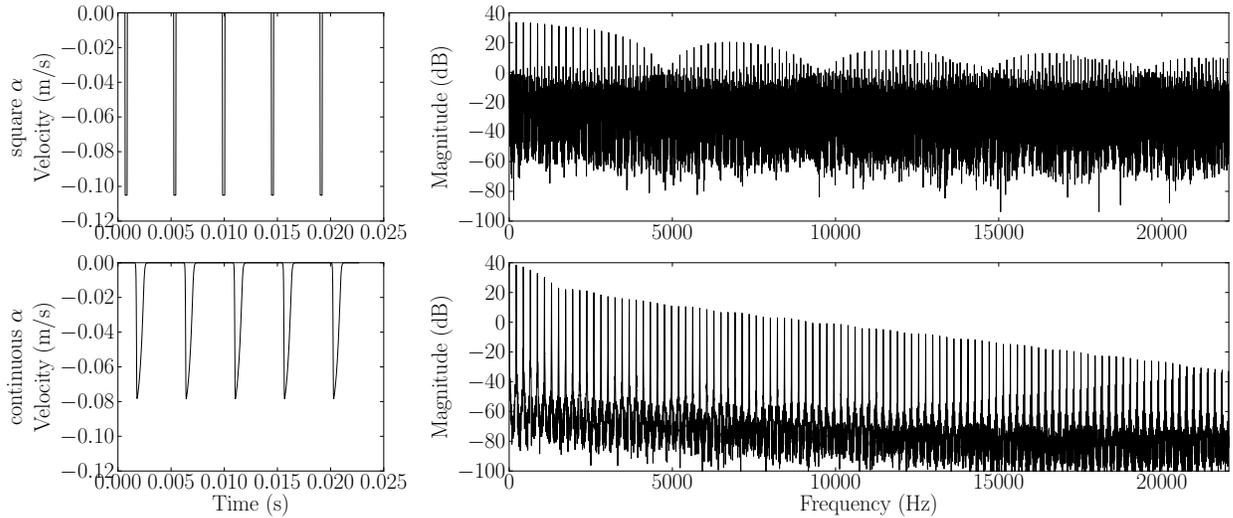
Normal force was held constant at 1.0 N , the fundamental frequency F_0 set to 220 Hz (open A3 cello string), and the value of z_{max} was 0.5 mm for all cases. The `DISTPLUCK` simulations used a smooth α with $n = 40$. For parameters affecting tension restoring force, $\beta = 0.87$, and $\lambda = 2.4 \times 10^{-3}$, giving $\gamma = 0.1$.¹ Good results were achieved with $\gamma \approx 0.07$ to 1.0 or so—the exact value does not seem to be critical, though small values produce sliding friction instead of oscillation, and high values cause the simulation to diverge.

Figure 4.7 shows that the presence of the H-A model in both the `TWO-POINT` and `DISTPLUCK` cases has the effect of delaying the attack onset. This is because with an initial $x_s = b = 0$, then b must reach z_{max} before anything significant happens. As mentioned, this helps avoid spontaneous noise due to small changes in the end effector position signal when the handle is not visibly moving. Therefore `DISTPLUCK` shares this desired characteristic with the previous `TWO-POINT` model. The z_{max} region is kept small so that the user has the impression that it responds quickly to direction reversal.

Another property is that the `DISTPLUCK` model does not exhibit a large degree of DC bias, as compared to `TWOPOINT`. This is due to the fact that the string position is

¹Unfortunately, the value of λ was wrong by a factor of the sample period in [120] due to a missing Δt in eq. 4.9; the correct value is given here. However, γ was correct, therefore the simulations were unaffected by this mistake both here and in that paper.

Incoming pulse velocity:



String velocity:

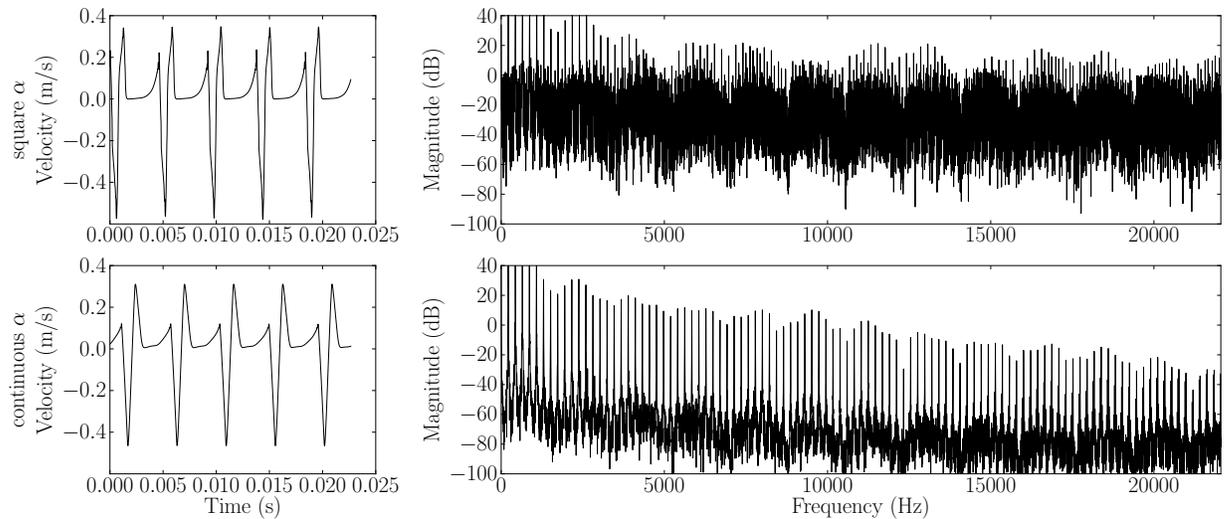


Fig. 4.6 Spectra for velocity pulses (top) and resulting string velocity (bottom) during the steady state signal in two configurations of the DISTPLUCK model, $F_0=220$ Hz, simulated in clean conditions. It can be seen that the continuous α (bottom) suffers significantly less from aliasing, as seen by comparing the inharmonic partials apparent in the square α spectrum due to repeated fold-over.

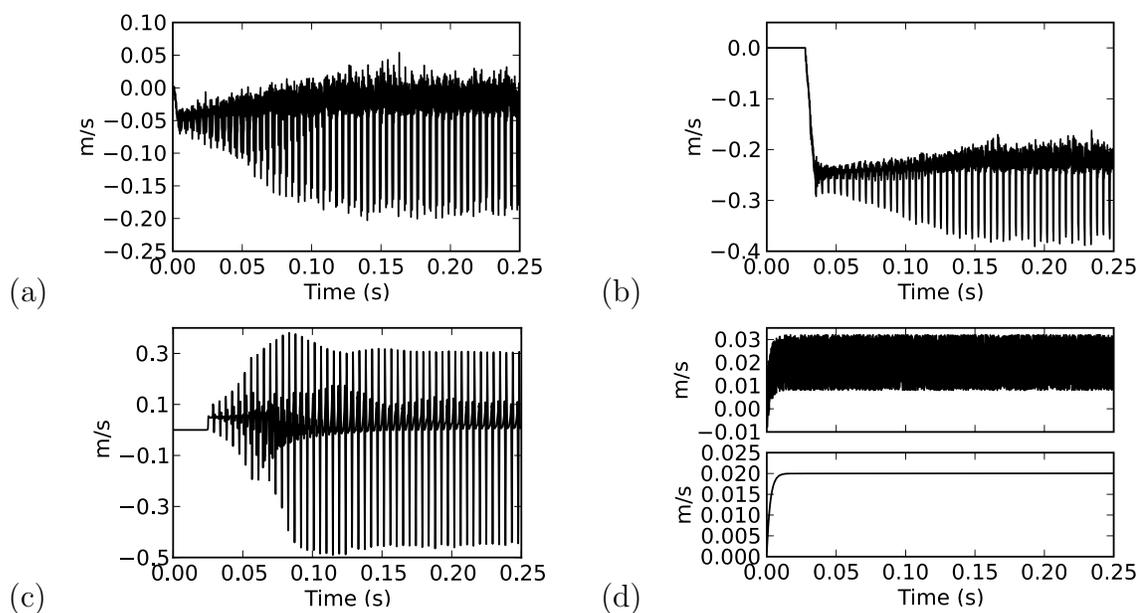


Fig. 4.7 .

String velocity, v_s , in noisy condition for models (a) BOWED, (b) TWOPOINT, (c) DISTPLUCK. (d) Simulation of bow velocity, v_s , from backward difference, noisy condition above, no noise below.

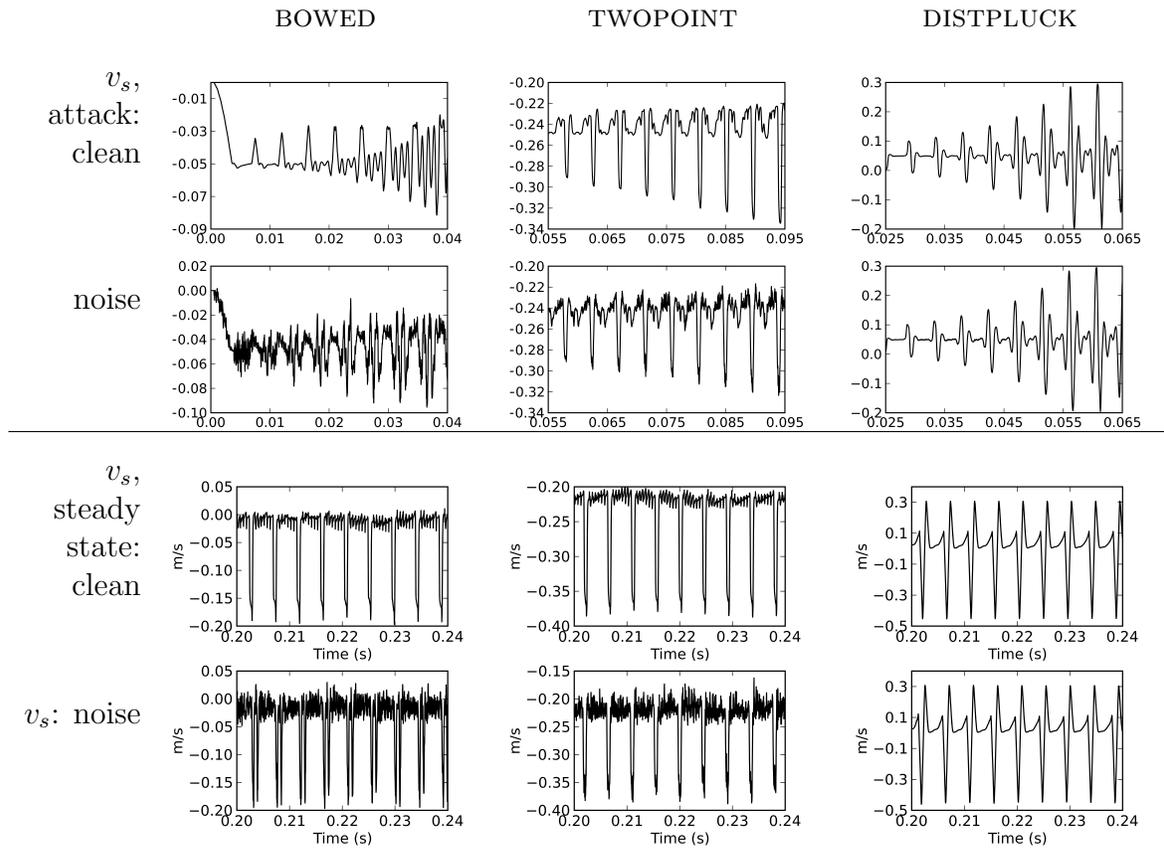


Fig. 4.8 String velocity v_s during the attack and steady-state of each model for clean and noise conditions. The input velocity is the same as that given in Fig. 4.7d.

constrained to the bow position, and the harder the constraint, the less offset is imposed—the discontinuous choice for α creates zero bias. For smooth α , the bias disappears after the attack reaches steady state: the resonator input between plucks is zero, whereas the TWOPOINT centers the oscillation around \dot{b} .

In general, DC-bias is not considered to be a problem because it would be filtered out by the instrument body, which is not modeled here. However, it does demonstrate a difference between the models, in that TWOPOINT continuously transfers the bow velocity to the string during stick, whereas DISTPLUCK ignores velocity transfer during stick and only simulates impulses that occur at release time.

In Fig. 4.8, notice the presence of noise in the steady state of BOWED and TWO-POINT as evidenced by the clear differences between clean and noisy conditions. In comparison, the string velocity for DISTPLUCK is almost identical in both conditions.

In BOWED and TWO-POINT there is also some amount of high-frequency vibration between impulses even in the clean condition that is not present in DISTPLUCK. This is circulation in the shorter of the two delay line portions due to reflection at the bow-string junction during stick, which is not modeled in DISTPLUCK: since v_r does not include a term to subtract propagated velocity between the two parts of the waveguide, string energy is always transmitted.

4.3.3 Online simulations

Fig. 4.9 shows a recording of a bow stroke using DISTPLUCK on a haptic device. The algorithm was executed at 30 kHz, and all parameters were the same as described in sec. 4.3.2. The fourth and fifth rows of Fig. 4.9 show the substantial difference in signal-to-noise ratio between the detected position and a backward-difference velocity estimate. Of course a better velocity estimator could be used, but we show the behaviour compared to these worst-case conditions of velocity estimation to demonstrate that the model output (string velocity) in DISTPLUCK is clean regardless of the velocity estimator, since velocity information here is essentially absent.

A phenomenon that became evident in online simulations which cannot be easily seen graphically is that there is a slight increase in pitch when the bow is lifted and the string left to resonate on its own. It seems that, when driven, impulses occur slightly late as compared to the natural string frequency, therefore producing a moderately lower tuning

by approximately 0.5 to 1.5 Hz. This is no doubt an artifact due to position-driven, rather than time-driven, impulse generation. Correcting for this is a topic still under investigation.

4.3.4 Conclusion

A model for haptic interaction with a bowed string was described that makes use of forces derived from a position-constraint friction algorithm. This avoids the necessity of deriving a velocity signal from measured position. Clean impulses are generated based on relative displacement, leading to stick-slip behaviour. Off-line simulations and measured results confirm that the model exhibits the desired properties.

Limitations include the lack of reflection at the bow-string junction, and a small but audible change in pitch when the bow is lifted off the string. Both of these issues are currently under investigation and will be the subject of future work.

We note also that there is a clear lack of audio quality when the rounded choice for α is used, which is difficult to account for—the model seems to sound much better in the squared α configuration, despite the nicer anti-aliasing properties of the rounded configuration. We suspect that this is likely related to the same tuning problem referred to above, where the string is driven by “late” impulses due to the position-based nature of the events. A continuous α leads to more variation in the oscillation frequency, which makes the string sound like its tuning is being modulated within a small range.

More generally, an issue we have with the DISTPLUCK approach is that our solution for adjusting the friction anchor point to correspond with string motion was highly model-specific. Therefore, it is not clear whether this approach is applicable to dynamic models in general.

In the case of DISTPLUCK, we were able to identify an aspect of the model, bow-string tension, which could be exploited to modulate the friction behaviour. However, this notion of tension is inspired by physics, but is not rigorously physical; for example, the restoring force due to tension is used to derive the amplitude of slip velocity, however in reality this force is more complex since the string is not static. Nonetheless, this calculation is needed since it was clear from testing that the amplitude must be adjusted according to the fundamental frequency—some amplitudes that were successful for 220 Hz strings led to instability for 110 Hz.

We believe that it should be possible to find a similar force in other dynamic models

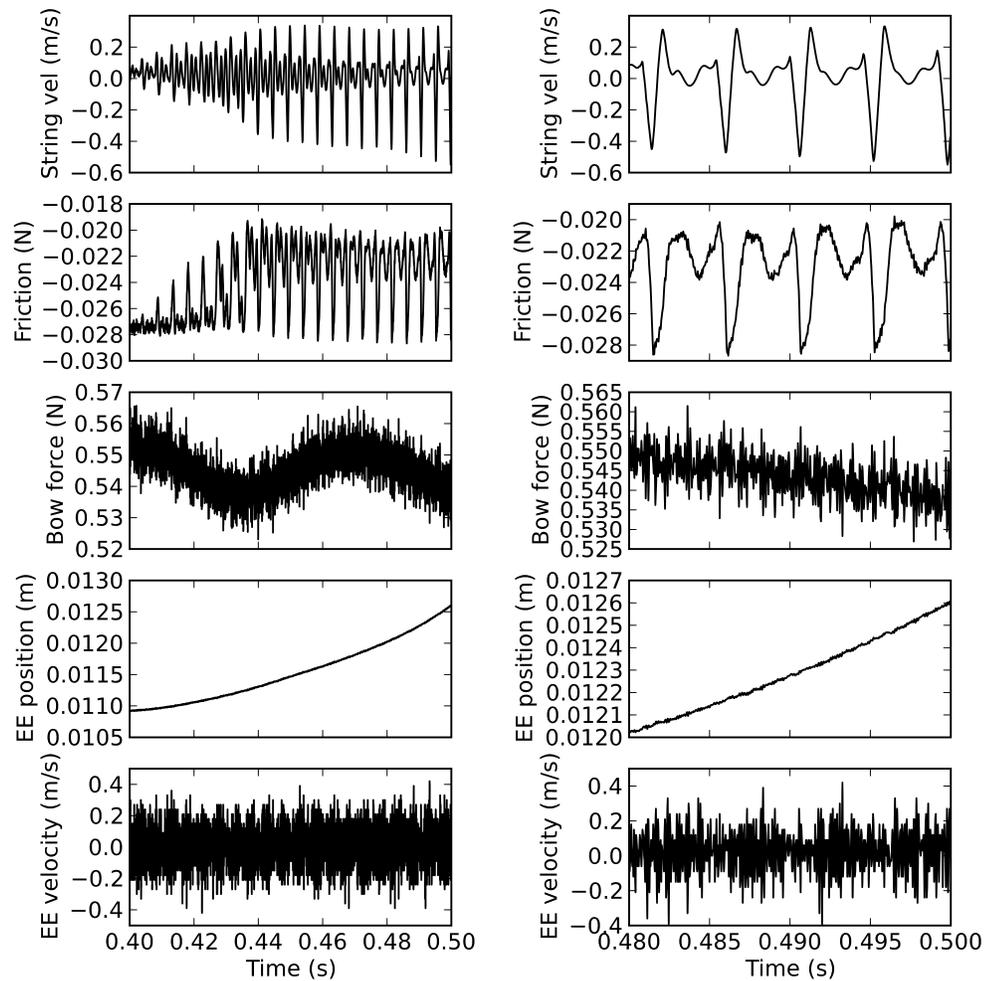


Fig. 4.9 Two zoom levels of a recording of DISTPLUCK running at 30 kHz on a haptic device. From top to bottom, the plots are: string velocity v_s , friction force, normal force, horizontal position of end effector, and velocity estimated by backward difference.

that exhibit stick-slip interaction—one that relates sticking tension between the end effector position and the model contact location—which can be used to control friction force; for now, this remains conjecture.

That said, this method could be considered among techniques of *haptic synthesis* [54], a term coined to refer to haptic rendering methods, analogous to audio synthesis methods based on signal processing such as frequency modulation, which produce physically ecological stimuli regardless of whether their internal details are rigorously physical in nature.

Chapter 5

Improvement by measurement and estimation

In the previous chapter, we described a method to drive a digital waveguide string model using a position-based friction, thereby avoiding the need to differentiate. For reasons outlined in that chapter's conclusion, we would like to explore the alternative means of improvement, consisting of improving the quality of the velocity estimate in a velocity-driven model. This can be approached by two main avenues: improved estimation algorithms, and improved sensing apparatus. Additionally, we can combine these approaches via sensor fusion methods, using sensor input to inform intelligent signal estimation.

As we have noted in other chapters, a key feature we require is to reduce estimation noise without introducing detrimental delay, which is unfortunately a feature of a very common approach to velocity estimation, the low-pass filter. However, there exist several methods, described below, that can improve estimates of the differential while decreasing the imposed delay. At least two of these have been previously proposed for force feedback interaction, and improvements to the impedance range of a damped virtual wall have been shown to be possible in each case due to the reduced delay.

On the other hand, it is reasonable to believe that appropriate sensing methods may further improve on these results. The reason is that any purely position-derived velocity is naturally inclined to delay. Already, position is delayed by one time step relative to the force command signal, so velocity, since it must take into account previous position samples, is delayed by at least 2 time steps [57].

More generally, for discrete position-controlled systems it is necessary to consider the Courant-Friedrichs-Lewy condition, which says that for explicit finite difference schemes, velocity may only be known within a quantum defined by time and space resolution:

$$\Delta > v_C T, v_C < \frac{\Delta}{T}. \quad (5.1)$$

Thus, sampling faster may potentially improve velocity resolution, but only if position resolution is sufficient [24]. Sensing higher derivatives directly may therefore allow for less delay due to these fundamental issues, since more information can be inferred about the same instant in time.

With the variety of sensing, estimation, and fusion methods, there are in fact a plethora of options for improving the sampled velocity signal used to drive the acoustic model. What is needed, therefore, is a way to compare them so as to help choose an appropriate method for our application. A complication is that many of the proposed methods have parameters that can be adjusted, making comparison difficult: it can be invalid to conclude that one method is superior than another, if it may be simply a matter of parameter tuning. Therefore we took the approach of performing parameter optimisation on each method according to criteria related to performance in terms of noise and delay, compared these off-line results across methods, and finally evaluated each technique using on-line subject-based evaluation. In this chapter we discuss the sensing apparatus used in the remainder of this dissertation, followed by a brief description of each estimation and fusion method selected for study; numerical optimisation and on-line evaluation are described in chapters 6 and 7 respectively.

Before proceeding, however, we first describe the context in which velocity estimators will be tested, by giving details on the velocity-coupled bowed string model we used during recording and testing in the next chapter.

5.1 Velocity-coupled bowed string

In order to test velocity-based bowed string interaction, it was necessary to choose a bidirectional coupling between the digital waveguide method [123], described in Chapter 1, and the force feedback device's end effector velocity and force signals.

As opposed to the position-dependent friction method described in Chapter 4, here

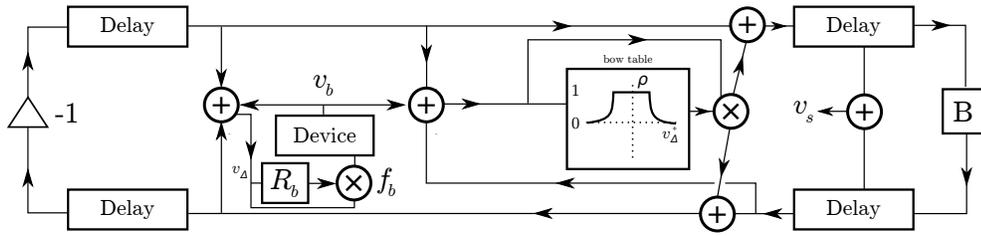


Fig. 5.1 Bowed string digital waveguide from Smith [123], enhanced with a string-on-bow force output to provide haptic feedback. The *Device* block represents the impedance haptic device, which outputs estimated velocity and takes a force command as input.

we wished to use a velocity-based bowed string interface involving physically-based force feedback from the bow-string junction. As mentioned in 2.2.1, Berdahl et al. [13] proposed two methods for doing this: the *explicit* and *implicit* junctions.

We found that our implementation of the explicit junction technique was highly unsatisfactory. Although the sound output was as expected, the force feedback was either too weak to feel, or very quickly became unstable as force gain was increased.

However, while pursuing an improved implementation, we derived a bow-string junction based on [125] that was later found to be equivalent to the implicit junction proposed by Berdahl. Our own derivation is described below, and we draw the correspondence with Berdahl’s work following this.

5.1.1 Bow force from the digital waveguide

Consider, from Smith [123],

$$F_b = -R_b(v_\Delta) \cdot v_\Delta \quad (5.2)$$

$$v_\Delta = v_b - v_s \quad (5.3)$$

$$= v_b - (v_{s,l}^+ + v_{s,l}^-), \quad (5.4)$$

where F_b is the force on the bow, expressed as the effect of the velocity differential on bow impedance R_b .¹

Recall from our description of the bowed string in Chapter 1, the relations between

¹Smith [123] states $F_b = R_b(v_\Delta) \cdot v_\Delta$, which has a different sign. We use the opposite frame of reference, to match the sign of device feedback forces, which shall be made clear below in section 5.1.2.

incoming and outgoing traveling velocities at the junction, (eqs. 1.3 and 1.4,) were,

$$v_{s,r}^- = v_{s,l}^+ + \rho(v_{\Delta}^+) \cdot v_{\Delta}^+ \quad (5.5)$$

$$v_{s,l}^- = v_{s,r}^+ + \rho(v_{\Delta}^+) \cdot v_{\Delta}^+. \quad (5.6)$$

Since we must calculate outgoing string velocity, the value for $v_{s,l}^-$ is available without delay, and it is possible, substituting eq. 5.6 into 5.4, to directly determine the outgoing bow force based on bow impedance using transmission coefficient ρ :

$$v_{\Delta} = v_b - (v_{s,l}^+ + v_{s,r}^+ + \rho(v_{\Delta}^+) \cdot v_{\Delta}^+) \quad (5.7)$$

$$= v_{\Delta}^+ - \rho(v_{\Delta}^+) \cdot v_{\Delta}^+. \quad (5.8)$$

Therefore direct calculation of F_b is possible. This is expressed in Fig. 5.1, a modification of Smith's bowed string block diagram where the device appears as a source and sink for bow velocity and force. Note that the device block represents a unit delay, since feedback force is calculated in response to the current velocity measurement, and zero-order-held until the next time step.

Bow pressure p , the second variable for bow impedance R_b and transmission coefficient ρ is derived from a vertical spring model, and is not described in this diagram. In practice we used a 1-DOF configuration for the device and held bow pressure constant at $p = 0.1$, where the units are a normalized ratio as employed by the STK *Bowed* object. We also used STK's default β -ratio of 0.13, the ratio of the position of the bowing point along the length of the string.

Implementation-wise, this formulation differs from the original in that it is necessary to use two look-up tables: one for ρ and a corresponding one for R_b , however it is still quite efficient. Nonetheless, since it depends directly on bow velocity, it is susceptible to noise in the device velocity signal.

Of course, R_b must be designed to match ρ . We fitted the width of R_b 's central viscous portion to correspond with the corner locations of ρ as they varied with normalized pressure p , see Fig. 5.2. The maximum point of $R(v_{\Delta})$ was designed to coincide with the corners of $\rho(v_{\Delta})$. This tends to give an equalized slope for the viscous portion across all p .

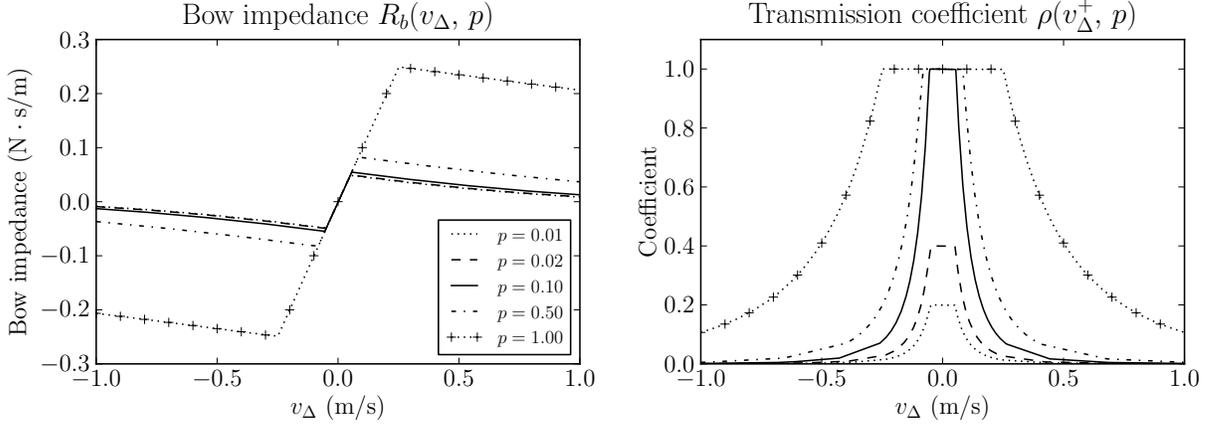


Fig. 5.2 Bow impedance and transmission coefficient for several values of vertical bow pressure. Here, bow pressure p has a dimensionless normalized unit between 0 and 1 as used by the STK software. For 2-DOF interaction, we scaled down the transmission coefficient when $p < 0.05$ to allow complete disconnection, seen here where coefficients for $p = 0.02$ and $p = 0.01$ are less than 1, however this was not used in our tests since we held $p = 0.1$ constant.

5.1.2 Correspondance to the implicit junction

This derivation in fact corresponds with the implicit junction proposed by Berdahl et al. [13].

In that work, a general waveguide coupling approach is to specify a spring k and damper R for coupling device position x_d and estimated velocity \hat{v}_d with model estimated position \hat{x}_s and velocity v_s ,

$$F_s = k(x_d - \hat{x}_s) + R(\hat{v}_d - v_s) = -F_d \quad (5.9)$$

The impedance relation between string incoming traveling wave velocities a_L , b_R , string velocity, and string wave impedance R_0 is,

$$v_s = a_L + b_R + \frac{F_s}{2R_0}. \quad (5.10)$$

Berdahl et al. [13] suggested using a leaky integrator with pole $p < 1$ as position estimator,

$$\dot{\hat{x}}_s = -p\hat{x}_s + v_s \quad (5.11)$$

Substituting 5.10 and 5.11 into 5.9 and solving for F_s ,

$$F_s = \frac{2R_0}{2R_0 + R} (k(x_d - x_s) - R(b_R + a_L - v_d)) \quad (5.12)$$

Discretizing by forward Euler,

$$\hat{x}_s[n] = \hat{x}_s[n-1] + \dot{\hat{x}}_s[n-1] \frac{1}{f_s} \quad (5.13)$$

$$\dot{\hat{x}}_s[n] = \frac{f_s - p}{f_s} \dot{\hat{x}}_s[n-1] + v_s[n] \frac{1}{f_s} \quad (5.14)$$

and substituting $p' = \frac{f_s - p}{f_s}$ gives,

$$F_s[n] = \frac{2R_0}{2R_0 + R} \left(k(x_d[n] - p' \hat{x}_s[n-1]) - v_s[n] \frac{k}{f_s} - R(a_L[n] + b_R[n] - v_d[n]) \right) \quad (5.15)$$

$$= \frac{2R_0}{2R_0 + R + k/f_s} (k(x_d[n] - p' x_s[n-1]) + Rv_d[n] - (R + k/f_s)(a_L[n] + b_R[n])) \quad (5.16)$$

where eq. 5.16 is the discrete string force relation from [13].

For bowing, a time-varying damper R is used, and the spring is set $k = 0$. This gives a simpler relation with no need for a position estimator,

$$F_s = \frac{2R_0 R}{2R_0 + R} (v_b - a_L - b_R) \quad (5.17)$$

Let us now change to the bowed string notation from the bow point of view, using $R = R_b(v_\Delta)$, $R_0 = R_s$, $a_L = v_{s,l}^+$, $b_R = v_{s,r}^+$, and bow velocity is considered the device velocity $v_d = v_b$.

$$F_b = -\frac{2R_s R_b(v_\Delta)}{2R_s + R_b(v_\Delta)} (v_b - (v_{s,l}^+ + v_{s,r}^+)) \quad (5.18)$$

Recall some definitions from Smith [125],

$$v_{\Delta}^{+} = v_b - (v_{s,l}^{+} + v_{s,r}^{+}) \quad (5.19)$$

$$v_{\Delta} = v_b - v_s = v_b - (v_{s,l}^{+} + v_{s,l}^{-}) \quad (5.20)$$

$$\rho(v_{\Delta}^{+}) = \frac{r(v_{\Delta})}{1 + r(v_{\Delta})}. \quad (5.21)$$

Breaking from [125], let the impedance ratio r be,

$$r(v_{\Delta}) = \frac{R_b(v_{\Delta})}{2R_s}, \quad (5.22)$$

then after some manipulation, F_b can be determined as,

$$F_b = -\frac{2R_s R_b(v_{\Delta})}{2R_s + R_b(v_{\Delta})} v_{\Delta}^{+} \quad (5.23)$$

$$= -R_b(v_{\Delta}) \left(v_{\Delta}^{+} - \frac{R_b(v_{\Delta})}{2R_s + R_b(v_{\Delta})} \cdot v_{\Delta}^{+} \right) \quad (5.24)$$

$$= -R_b(v_{\Delta}) (v_{\Delta}^{+} - \rho(v_{\Delta}^{+}) \cdot v_{\Delta}^{+}) \quad (5.25)$$

$$= -R_b(v_{\Delta}) \cdot v_{\Delta}, \quad (5.26)$$

matching eq. 5.2, above. This shows that the two formulations are equivalent.

If we use the impedance ratio from [125], we get,

$$r(v_{\Delta}) = \frac{R_b(v_{\Delta})}{4R_s} \quad (5.27)$$

$$F_b = -\frac{4R_s R_b(v_{\Delta})}{4R_s + R_b(v_{\Delta})} v_{\Delta}^{+}. \quad (5.28)$$

Therefore these methods differ by a factor of two in the assumed impedance ratio, but this is not a highly significant factor and may reflect different choices of material properties.

5.2 Sensors

5.2.1 Position

Most commonly, force feedback devices feature position sensing so as to enable display of position-dependent effects such as virtual walls, the building blocks of many types of interaction scenarios. Since position can be differentiated to derive velocity, it is a natural lowest-common-denominator sensing apparatus which can additionally be used to support virtual damping when needed, therefore position sensors are the go-to solution for commercial devices which are expected to operate at 1 kHz.

Position sensors can be in a rotary or linear configuration, and generally-speaking can take the form of either analog voltage sensors, which provide a DC voltage with some monotonic relation to displacement, or digital encoders, which are usually optical non-contact sensors based on the modulation of light by a movable grating. The former may be referred to as “fixed-time” sensors, and the latter as “fixed-position,” since analog sensors are usually sampled at discrete time intervals, while encoders provide an electrical trigger whenever a discrete position change is detected [17].

The fixed-time approach is more sensitive to electrical noise, though fixed-position sensing experiences delay between measurements at low velocities. Often for simulation purposes the fixed-position approach is time-quantized in order to allow a regular time step for integrators, further adding to inaccuracy. However, using programmable digital hardware it is actually possible to design haptic loops which react nearly instantaneously to position changes by taking an asynchronous computing approach [130].

Analog voltage sensors may use many different means to detect displacement, such as resistance or capacitance of an electrical signal, or amplitude of optical or acoustic signals. The TGR features a linear variable differential transformer (LVDT), a position sensing technique based on electrical inductance: an alternating current is induced by the central coil in a conductive bar which moves within 3 coils physically in series; the two secondary coils on either side are connected in opposition, so that their induced voltage from the bar is amplified differentially according to displacement. Demodulation is then used to determine a DC voltage which is linearly proportional to displacement [86]. The LVDT is thus a fixed-time sensing technique.

In the Ergon-X system, this voltage is measured and quantized by the DSP system using a 16-bit analog-to-digital converter, giving ideally a resolution of 0.336 μm for the

2.2 cm available displacement if the full sensor range is used. In practice, we measured an RMS amplitude of 15 μm in the sampled analog signal when at rest.

5.2.2 Velocity

Velocity can be derived from the fixed-time method by measuring the change in position over time. For the fixed-position approach, there are two basic methods: to measure the time interval between events, requiring a high resolution clock, and to measure the number of events over time, requiring a high resolution encoder; these can be combined, for example by counting the amount of time over a set number of events [88]. Although timer-based methods can provide very good resolution at low velocity, the time between individual measurements also increases, which can be detrimental in haptics. Note especially that zero velocity cannot be detected in this manner, except by enforcing a maximum timer count. Observers and other model-based methods can be used to estimate inter-event state [60], but these encoder-related estimation techniques were not explored in this thesis since we used a fixed-time sensor. However, encoder position measurements may be applied to the estimation techniques discussed below.

There are comparatively fewer ways to measure velocity directly—that is, by measuring a physical signal that is proportional to instantaneous velocity. One is by exploiting the Doppler effect [94]: an oscillating acoustic or optical signal will change frequency if it is emitted from a moving source, since change in the path length causes an elongation or compression of phase. This can be accomplished by a moving emitter, or by a moving reflector. The frequency change is proportional to velocity, so that if frequency is detected, velocity can be inferred. Detecting frequency can be accomplished by a timer circuit or by interferometry; trade-offs in the requirements of timing circuitry and the base frequency exist, and while heterodyne interferometry can be used to improve precision, small frequential differences may take time to detect. For example, the beating frequency of a 1 Hz difference takes at least one second to detect, meaning that precision and delay are a trade-off with this method. Note also that ultrasound signals may be too low-speed depending on the application, and laser interferometry is an expensive solution that is better-suited to long range sensing.

For rotary workspaces, a vibrating structure gyroscope may be an interesting option [92]. An inertia vibrates in one axis, implemented as e.g. a driven piezoelectric material,

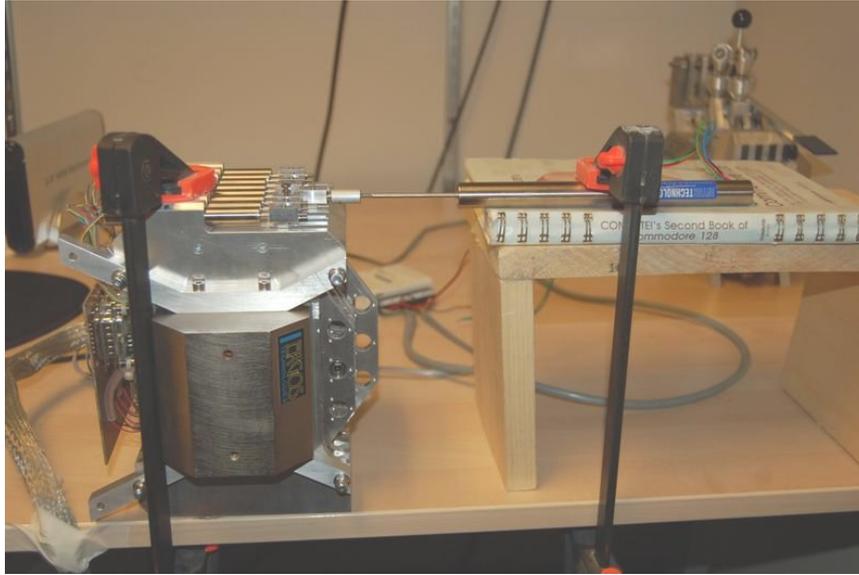


Fig. 5.3 The TGR device in a horizontal 1-DOF configuration, coupled to the Trans-Tek Series 100 LVT tachometer.

or embedded in silicon in an integrated circuit (MEMS). The vibrating object tends to preserve the plane of vibration, and the Coriolis force leads to vibrations in the orthogonal axis. Change in angular momentum, and therefore velocity, can thus be detected. Laser gyroscopes are also an option, but use a different physical principle based on the shortening and lengthening of distance traveled during counterrotation of a circular optical path, which can be detected by homodyne interferometry.

Finally, another method of detecting velocity is magnetic induction: by Lenz' law, a coil containing a moving magnet will have a voltage induced to counter the change in magnetic flux. Since this generated voltage is linearly proportional to this change, it is proportional to the velocity of the magnet. In industry, a linear velocity transducer (LVT) consists of a bar-shaped permanent magnet inside a regular coil, which takes advantage of this physical law to provide a region with a linear velocity-voltage relationship [92]. It is restricted in range, but this does not pose a problem for our use, where the workspace is only a few centimeters. This latter method is therefore chosen as a “ground truth” for comparison purposes in this work.

We used the Series 100 LVT (model 0112-0000) from Trans-Tek, which we will refer to as “the LVT” or “the tachometer” throughout this work. From the datasheet, it has a working linear range of 5.0 cm (1.97"), an electrical impedance of 19 k Ω and 2.9 H, and

a frequency response of at least 500 Hz. The moving magnet has a mass of 15 g. We measured a noise amplitude of 0.115 mm/s after analog-to-digital conversion when at rest.

A photo of the device coupled to the velocity sensor can be seen in Fig. 5.3.

5.2.3 Acceleration

Alternatively velocity can be integrated from the next higher derivative, acceleration. In comparison with differentiation, integration has the opposite effect on noise, improving rather than reducing the signal-to-noise ratio. However, because accelerometers are ungrounded devices, this use of accelerometers does have the problem of an unknown initial value, and additionally numerical error and noise accumulates as a low-frequency drift. These problems can be summarized as a low-frequency error which needs to be removed by the use of a high-pass filter. Therefore accelerometers need to be combined with a source of accurate low-frequency information. This can be derived from a position signal, already available as a standard component on the device; we describe this in more detail in 5.4.1, below.

For this work, we used the model 352C22 accelerometer from PCB Piezotronics.² It is a high-performance accelerometer designed for acoustic measurements. It weighs 0.5 grams, has a measurement range of ± 4900 m/s² (± 500 gravities), and a frequency range up to 10 kHz. These piezo-accelerometers take advantage of the piezoelectric effect, where deformation of the material produces a voltage. By adding a known inertia to the piezoelectric material, acceleration can be derived based on the deformation created by the inertia's opposition to changes in the enclosure's velocity.

This accelerometer was attached to the device using a waxy substance provided with it, intended for this purpose. We measured a constant RMS noise level of 0.22 m/s² (0.022 gravities) from this sensor when at rest.

We also tested using a much less expensive ADXL-203 sensor from Analog Devices, having a range up to 2.5 kHz. This accelerometer is used in consumer electronics and is widely available. Unfortunately the results were not good due to saturation, therefore we have not presented this data.

Another possibility for extended sensing could be to use force sensors to detect the force vector that the user is applying to the end effector. By combining this with commanded

²http://pcb.com/spec_sheet.asp?model=352C22

force and known inertia and friction properties, an estimate of acceleration, and therefore velocity, could be derived. This idea was not tested in the current work.

5.3 Differentiators

This section will describe some methods for estimating velocity based purely on position. These methods fall into two categories: estimators and observers. An estimator is a function combining current and past observations to generate a best guess of some derivable value—in our case, velocity from position. An observer is a system model driven to follow observed signals and whose states can be used to determine unobservable internal system states [101]. In other words, an estimator asks, “given some set of observations, what is the most likely state currently?,” while an observer asks, “if the given observations are modeled, what must the internal state necessarily be?”

In both cases, the goal is to derive an estimate of the desired signal as exactly and instantaneously as possible. A “perfect” estimator or observer for velocity would reflect the exact position derivative at the precise moment of sampling. In practise both estimators and observers typically incur some delay, for example to ensure smoothing of noisy measurements. Observers with higher gain will generally have faster response times, but may accentuate noise by reacting quickly to outliers. Therefore some designs may perform better than others. In chapter 6 we will evaluate the performance of some estimators and observers with regards to their performance as a delay/error trade-off, and compare these to methods which also incorporate higher-order measurements, described in section 5.4.

5.3.1 Backward difference

The simplest method of differentiating is by first-order backward difference, however it is also considered the least well-performing in terms of noise characteristics. Defining x_k as the signal $x(t)$ at time $t = hk$, where h is the sample period, it approximates the differential $\frac{dx}{dt}$ as,

$$\hat{v}_k = \frac{x_k - x_{k-1}}{h}, \quad (5.29)$$

where \hat{v}_k is the velocity estimate for sample k .

Since this defines velocity as the change in position over time, it effectively calculates

the average velocity over the sample time [17]. It follows that more precision can be had at the expense of time by taking a longer average, i.e.,

$$\hat{v}_k = \frac{x_k - x_{k-z}}{h}, \quad (5.30)$$

where $z \geq 1$ is some past sample of position. By the same logic, decreasing the time period improves response time but *worsens* the precision of the result in the presence of noise.

One solution, proposed e.g. in [31], is to use averaging or a low-pass filter on the \hat{v}_k estimate, however this solution is very similar in spirit to lengthening the differential window, taking several estimates into account rather than fundamentally improving the original estimate.

Note also that this averaging effect means that even with $z = 1$, it is not the instantaneous velocity at the most recent sample time being determined, but rather the velocity at the average time between the two samples. Therefore this method is naturally delayed by at least half a sample period.

5.3.2 Least squares fit

The averaging approach of backward difference follows from the fact that a linear fit is applied to a sample window and the velocity is taken as the slope, which is assumed constant over the window. It follows that an improvement can be had by estimating a higher-order fit, and taking the derivative at the most recent sample time.

Brown et al. [17] proposed the use of a linear least squares estimator for this purpose, expressed as a set of finite impulse response (FIR) filter coefficients. The linear least squares fit, the most common technique in data fitting, is a method to find the best N -order polynomial through a set of M points such that the sum of the squares of the error is minimized. From [17], the FIR coefficients can be found by assuming an N -order estimate of position,

$$\hat{x}_k = c_0 + c_1 t_k + c_2 t_k^2 + \cdots + c_N t_k^N, \quad (5.31)$$

where t_k is the relative time at sample k .

The differential is then,

$$\frac{d\hat{x}_k}{dt} = c_1 + 2c_2 + \dots + Nc_N t_k^{N-1}. \quad (5.32)$$

Expanding eq. 5.31 to include the last M samples, we can express it as a linear combination in matrix form,

$$\hat{x} = Ac, \quad (5.33)$$

where \hat{x} contains the last M position measurements, c is set of $N+1$ polynomial coefficients, and A is a matrix containing N powers of t_i at row i . If we assume a sample period $h = 1$, leaving a correction to the real sample period for the end, then A is expressed as,

$$A = \begin{bmatrix} 1 & 1 & 1^2 & \dots & 1^N \\ 1 & 2 & 4 & \dots & 2^N \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & M & M^2 & \dots & M^N \end{bmatrix}. \quad (5.34)$$

According to standard least squares fit procedure, the coefficients that minimized the error are,

$$c = (A^T A)^{-1} A^T x = A^\dagger x. \quad (5.35)$$

Then a vector $\dot{q} = [0 \ 1 \ 2M \ 3M^2 \ \dots \ NM^{N-1}]$ can be used to take the derivative with respect to time, analogous to eq. 5.32,

$$\frac{d\hat{x}}{dt} = \dot{q}^T A^\dagger x = \dot{h}^T x = \dot{v}_k, \quad (5.36)$$

where \dot{h}^T represents a linear combination of previous samples, i.e. our FIR coefficients. Finally, the estimate \dot{v}_k is divided by the sample period, since we assumed $h = 1$ above.

In the next chapter, we evaluate this technique for values of $N = 1$ to 4, to see whether higher-order estimates might give better results. We expect that much higher order polynomials would re-introduce noise, since such an estimate would come closer to the sampled points rather than smoothing them, therefore it is likely that some low $N < 1$ might provide

the best compromise between noise rejection and temporal precision.

5.3.3 Adaptive windowing

Since the essential problem with backward difference is a trade-off between time delay and the quality of the velocity estimate, a method proposed for haptics is the first-order adaptive windowing filter (FOAW) [61]. This approach recognizes that during high-velocity periods, there is less noise in the differential approximation. This can be understood intuitively by realizing that noise is accentuated in differentiation due to subtraction of very close values. During slow movement, consecutive position samples are very close, and therefore subtracting them produces very small numbers, which are closer to the noise margin, creating a small signal-to-noise ratio (SNR). During faster movement, position samples are further apart, conversely increasing the SNR.

Meanwhile, in a closed loop system, it is more important to have good time resolution when the end effector is moving, which corresponds with the increased signal-to-noise ratio. During periods of little to no motion, longer averages may produce better estimates without penalty. Therefore, if the window can be shortened during periods of high velocity, and lengthened during periods of rest, a better overall estimate can be produced without sacrificing time resolution.

In the FOAW approach, the expected properties of the SNR are used to determine the window size. It is an iterative approach which performs several consecutive fits to the position samples, and evaluates each according to the expected noise margin. The longest-time fit which lies within the noise margin is selected as the most likely candidate for differentiation. The slope is then taken as the velocity. In pseudocode,

```

for i in 1..N,
     $\mathbf{w}_i = [x_{k-i} \dots x_k]$ 
     $\mathbf{L}_i = \text{line\_fit}(\mathbf{w}_i)$ 
    if  $\max(|\mathbf{w}_i - \mathbf{L}_i|) < \mathbf{e}$ :
        continue
    else
        return  $\text{slope}(\mathbf{L}_{i-1})$ 

```

for some a priori estimate of expected signal noise \mathbf{e} .

In fact, two alternatives for FOAW were proposed in [61]. In the *end-fit* method, `line_fit` is an efficient fit against the current sample x and the first window sample, x_{k-i} . In the *best-fit* method, `line_fit` is a least squares line fit. The *best-fit* method gives better results, but at the expense of computation time, since every sample in the window must be evaluated each time through the loop, for every time k .

In the next chapter, we optimise for \mathbf{N} and \mathbf{e} using the *best-fit* method.

Finally, a method for using a similar approach for acceleration estimation has been proposed, called the extended adaptive window method (EAW) Kilic et al. [65]. It works by similar principles, but performs a higher-order polynomial fit to each successive window size. We did not evaluate EAW in this work, since we were principally interested in velocity, however it suggests that similar ideas to the least squares method could be applied to the *best-fit* method, to evaluate the derivative at the sample time rather than taking the slope of the first-order fit over the window.

5.3.4 Levant’s differentiator

Finally, we turn our attention to an observer approach. A particular observer called a *2-sliding observer* was recently proposed for haptics applications [28]. The 2-sliding method as a differentiator was proposed by Levant [72] as a “robust” and “exact” differentiator, hence we refer to it as Levant’s differentiator, following [28].

Sliding mode control is designed to overcome model inaccuracies, often arising from uncertainties or linearising assumptions in complex processes. It is appropriate for use when information about the signal structure is unknown, which is the case for haptics since the presence of an operator adds a large degree of uncertainty to the behaviour of the signal. It employs high-frequency switching control to maintain the system around a given constraint in its state-space.

r -sliding mode, or higher-order sliding modes (HOSM), are able to maintain a constraint up to its r^{th} derivative [73]. Therefore 2-sliding mode control can use a two-stage process to impose finite-time convergence for the constraint $\sigma = \dot{\sigma} = 0$, where, for an observer w on the signal x ,

$$\sigma = w - x(t) \tag{5.37}$$

$$\dot{\sigma} = u - \dot{x}(t), \tag{5.38}$$

where u is an observer on the differential. If $\dot{w} = u$, then $u(t)$ can be taken as an estimate of $\dot{x}(t)$.

This observer is “robust” because it is guaranteed to converge on the correct estimate regardless of initial conditions, and it is “exact” because it can follow the trajectory exactly given only an upper bound on the differential, as opposed to e.g. high-gain observers which are only exact if the gain tends to infinity. Stated otherwise, the 2-sliding observer is not bound to a particular frequency band; it has the useful property of following the target signal without delay, making it a very good candidate for our application.

The limitation is that this property only holds if there is an upper bound on the higher-order differential—the maximum acceleration, in other words, must be specified a priori, because it provides a Lipschitz constant for the signal that guarantees differentiability. Given this condition, the switching gain can be selected to guarantee that the sliding mode is maintained. Note that this implies that the higher the expected acceleration, the larger the switching gain required, which can result in significant noise if large accelerations are expected.

Levant [72] gives the control laws for the differentiator as,

$$\dot{w} = u \tag{5.39}$$

$$u = u_1 - \lambda|w - x(t)|^{1/2} \operatorname{sgn}(w - x(t)), \tag{5.40}$$

$$\dot{u}_1 = -\alpha \operatorname{sgn}(w - x(t)), \tag{5.41}$$

where $\alpha, \lambda > 0$, and u_1 is an additional observer state.

Here, α is some proportion of C , and λ some proportion of \sqrt{C} , where $C > 0$ is the Lipschitz constant, which guarantees local differentiability of $\dot{x}(t)$ if its derivative stays under a limit, i.e. if the following inequality holds:

$$|\dot{x}(t_k) - \dot{x}(t_{k-1})| \leq C|t_k - t_{k-1}| \tag{5.42}$$

Recommended choices are [72],

$$\alpha = 1.1C \tag{5.43}$$

$$\lambda = \sqrt{C}, \tag{5.44}$$

which we use in our implementation, following [28].

Other choices are possible. Sufficient conditions for convergence of $u(t)$ to $\dot{x}(t)$ are [72],

$$\alpha > C \tag{5.45}$$

$$\lambda^2 \geq 4C \frac{\alpha + C}{\alpha - C}. \tag{5.46}$$

We refer the reader to [72] for derivation and proofs.

Chawda et al. [28] notes that a significant advantage for this technique compared to other velocity estimators is that its performance increases with sampling rate, as opposed to backward-difference techniques which worsen. This can be understood by noting in eq. 5.42 that both subtractions are smaller for smaller sample periods. However, as noted above, high accelerations may impose significant switching noise on the output, and we found this to be the case when taking into account sharp transients such as tapping the TGR end effector against the aluminum stops of its workspace. It is recommended in [72] to follow the differentiator with a low-pass filter, which is one condition that we test in the next chapter, however this implies re-imposing the delay and frequency response limitations of filtering. We therefore additionally evaluated this method as input for a Kalman filter, described in 5.4.2, below.

5.4 Fusion techniques

Sensor fusion is a name for the process of combining two or more sensor readings to produce a single physical measurement. After conversion to a common unit, sensor readings may provide redundant information in the frequency domain. Filters can therefore be used to select the most accurate and precise bands of each sensor’s frequency response, eliminating overlap. Such a set of filters is called a *complementary filter* [8].

One criticism is that a complementary filter throws out information that could be potentially used to improve the signal. Another approach is to make use of statistical maximization to estimate the most likely “true” signal value at a given time. One such technique that is often used for this purpose is the Kalman filter, which at each step updates a prediction based on a dynamic system model, and then corrects the prediction using an optimal combination of weighted error judgements based on the expected covariance of each input with the measurement error and the model uncertainty [14].

Although there exist other fusion techniques based on statistical likelihood, particularly in the area of machine learning, we have selected only the complementary filter and Kalman filter for comparison in this work. Since the selected sensors are well-matched to both of these approaches, we do not require the greater flexibility that comes for example with training-based approaches. Due to the optimality properties of the Kalman filter [14] and its already quite large design space, we also did not pursue other stochastic techniques. The rest of this chapter will describe these two techniques in more detail, and they will be tuned and evaluated in the next chapter, after a discussion on error metrics.

5.4.1 Complementary filter

The idea of combining two or more filters raises the question, which filters should we use? The choice should balance good stop-band rejection properties with a flat response at the cross-over and a fast step response free of resonance.

There are several filter choices available. A common choice for most filter applications is the Butterworth, which has extremely flat response in the pass-band [84]. The Bessel filter is similar, but has its emphasis on a linear phase response [84]. Although a 1st-order Butterworth (BW1) cross-over is quite flat, in practise we found that a 1st-order choice does not have strong enough stop-band rejection. The 2nd-order cases of both BW2 and Bessel-2 do suffer from cross-over problems. An alternative is known as the 2nd-order Linkwitz-Riley filter (LR2), which is simply composed of two cascaded BW1 filters, and has the rejection properties of BW2, but with a flat frequency response at the cross-over [75]. We did not explore elliptical and Chebychev filters in this work [84]. Similarly, we limited our evaluation to 2nd-order filters, since higher orders would introduce extra delay.

Finally, although the LR2 model promises better cross-over behaviour, some initial testing did not show a significant difference between BW2, Bessel-2, or LR2 filters. Therefore we used the most common choice, BW2, for this work. Digital filter kernels were generated using filter design tools available in the SciPy scientific programming platform [62].

5.4.2 Kalman filter

The Kalman filter is an observer which uses a state-space process model to predict the next system state, and evaluates the error between its prediction and the next measurements using a stochastic model of the measurements and model uncertainty. It is a wide topic that

allows for many design choices, and as there is a plethora of introductory material easily available that describes it in detail, only a brief description will be given here, summarized from Bishop and Welch [14], along with notes on the selection of parameters used in this work.

At each step, the Kalman process updates its state according to the model, and adds to this a weighted difference between predicted and actual measurements. The weighting of this residual (the “Kalman gain”, K) is dynamically updated based on the error covariance, determined by the given measurement and process covariance. These noise sources are assumed to have a normal, zero-centered Gaussian distribution. The error covariance is itself updated at each time step by the Kalman gain, reflecting the confidence that the filter has a correct result.

Using the notation in [14],

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \quad (5.47)$$

$$P_k^- = AP_{k-1}A^T + Q, \quad (5.48)$$

where A and B are the linear model coefficients, \hat{x}_k^- and \hat{x}_k are the predicted and estimated states, u_k is the command signal, P_k^- and P_k are the predicted and estimated error covariance, and Q is the provided process covariance.

We can see that the initial state prediction \hat{x}_k^- is updated according to a process model, as in any linear state space system. Next, this prediction is corrected by measurements, which have a given reliability expressed by another covariance matrix R .

The measurement reliability along with the predicted error covariance determines a gain K on the residual, which must minimize the error between the prediction-plus-residual and the true value. Discussion of this minimization is outside the current scope, but one form of K is given by,

$$K_k = \frac{P_k^- H^T}{HP_k^- H^T + R}, \quad (5.49)$$

where H is a linear transformation matrix between measurement space and state space—note that this implies we assume to model a linear system. Finally the state and error

covariance estimates can be updated based on measurements z_k as,

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \quad (5.50)$$

$$P_k = (I - K_k H)P_k^- . \quad (5.51)$$

Note that calculation of K_k requires a matrix inversion, but this can be done quite efficiently for a matrix of small dimensions as we require.

In our case we have a one-dimensional linear axis of movement, which affords quite a simple process model of a point with position, velocity and acceleration states, i.e. a double integrator. If this were a free-moving piston, we could predict acceleration by the previously commanded force; however, since a user will be loading the device and applying his own external disturbances, we cannot assume to predict acceleration and we leave it as a free, untrusted dimension in the process model. For the same reason, we set $B = 0$.

We therefore have \hat{x}_k and \hat{x}_k^- as 3-vectors of the form $[x \dot{x} \ddot{x}]^T$ and so A is a 3×3 matrix. Since we have a discrete system, A must express an update for \hat{x}_k^- over time τ , where time $t = k\tau$, and similarly Q must describe the discrete propagation of noise through the model. From [8, p. 274], if we have a 3rd-order system,

$$x_{k+1} = Fx_k + \Gamma v_k,$$

where v_k is the process noise, then,

$$F = \begin{bmatrix} 1 & \tau & \frac{\tau^2}{2} \\ 0 & 1 & \tau \\ 0 & 0 & 1 \end{bmatrix} \quad \Gamma = \begin{bmatrix} \frac{1}{2}\tau^2 \\ \tau \\ 1 \end{bmatrix}. \quad (5.52)$$

For (5.47) and (5.48) we set $A = F$, and,

$$Q = \Gamma\sigma^2\Gamma = \begin{bmatrix} \frac{1}{4}\tau^4 & \frac{1}{2}\tau^3 & \frac{1}{2}\tau^2 \\ \frac{1}{2}\tau^3 & \tau^2 & \tau \\ \tau^2 & \tau & 1 \end{bmatrix} \sigma^2, \quad (5.53)$$

where σ is the power spectral density of the process noise.

Finally, for matrices relating to measurement, we have two inputs: position and ac-

celeration with variance σ_p and σ_a respectively. If we assume that any necessary signal conditioning and conversion from voltage to physical units has already been performed, then,

$$H = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (5.54)$$

The measurement covariance R will depend on sensor performance and sampling noise:

$$R = \begin{bmatrix} \sigma_p & 0 \\ 0 & \sigma_a \end{bmatrix}. \quad (5.55)$$

We note that the Kalman filter has been applied to velocity estimation for applications in speed control. For example, Kim et al. [66] applied this method to the two estimates available from timer- and event-driven encoder count measurement. Bélanger et al. [10] used a time-varying Kalman filter to update state estimates at intervals driven by both the encoder and clock events.

Because the Kalman filter is considered a standard technology for state estimation, and because it can be considered an optimal estimator for combining multiple measurement sources as long as measurement noise can be said to have a normal distribution, we considered it an interesting candidate to compare against other proposed methods for force feedback control, particularly non-model-driven methods such as FOAW and the 2-sliding observer.

Although our use of a 1-DOF double-integrator linear model is a simplifying assumption, this is a common approach in the haptics literature as a first approximation to a manipulator model (e.g. [3, 16, 32, 38]). It is acknowledged that we could benefit from more accurate models, possibly involving non-linear dynamics, however such extensions are considered beyond the scope of this work.

5.4.3 Hybrid solutions

While we may expect excellent results from the Kalman filter approach due to its properties of optimality, this comes with some caveats. In particular, the Kalman filter expects measurements to have a normally-distributed, bias-free error. Unfortunately this is not strictly true for our accelerometer sensor. In general, most accelerometers on the market

have a frequency band that does not go to DC—in other words, they do not allow perfect drift-free position tracking by means of integration, but rather bias, e.g. due to gravity, must be removed by high-pass filtering. This in turn introduces a bias towards zero during extended moments of acceleration.

It stands to reason that the Kalman estimator may benefit from preprocessing that can shape measurement input to the expected form. To investigate this approach we constructed some hybrid methods by adding results from our non-linear velocity estimators to the acceleration measurements via double-differentiation before feeding the mixed data to the Kalman update. Although this should increase the error margin of measurements, since the Kalman estimator is optimally robust to noise we wished to investigate whether the benefit of removing bias might exceed the drawbacks of increasing the measurement variance.

We tried using the output of FOAW best-fit and Levant’s differentiator to double-differentiate position measurements and mixed these results with the accelerometer measurements by simple addition. Although double-differentiation results in considerable noise, we did not pre-filter the results, since this would lead to delay and defeat the purpose of the experiment; rather, we preferred to see how the Kalman estimator would handle the larger covariance.

Since in the cases of FOAW and Levant’s differentiator the estimation process is non-linear, we found that the integration actually reduces the noise beyond what could be achieved by performing the addition between estimated velocities. This is demonstrated in Fig. 5.4, which shows a simple addition between the integrated double-differentiation and the accelerometer signal.

This method is thus similar to a complementary filter but leverages the improved noise and delay performance of the non-linear differentiators, rather than using a delay-inducing low-pass filter over the position measurement.

5.5 Trade offs

In the next chapters, we optimise and evaluate the above algorithms for their performance in terms of noise rejection vs. time response. However, there are some other trade-offs that may be considered when choosing an estimator that we summarise here.

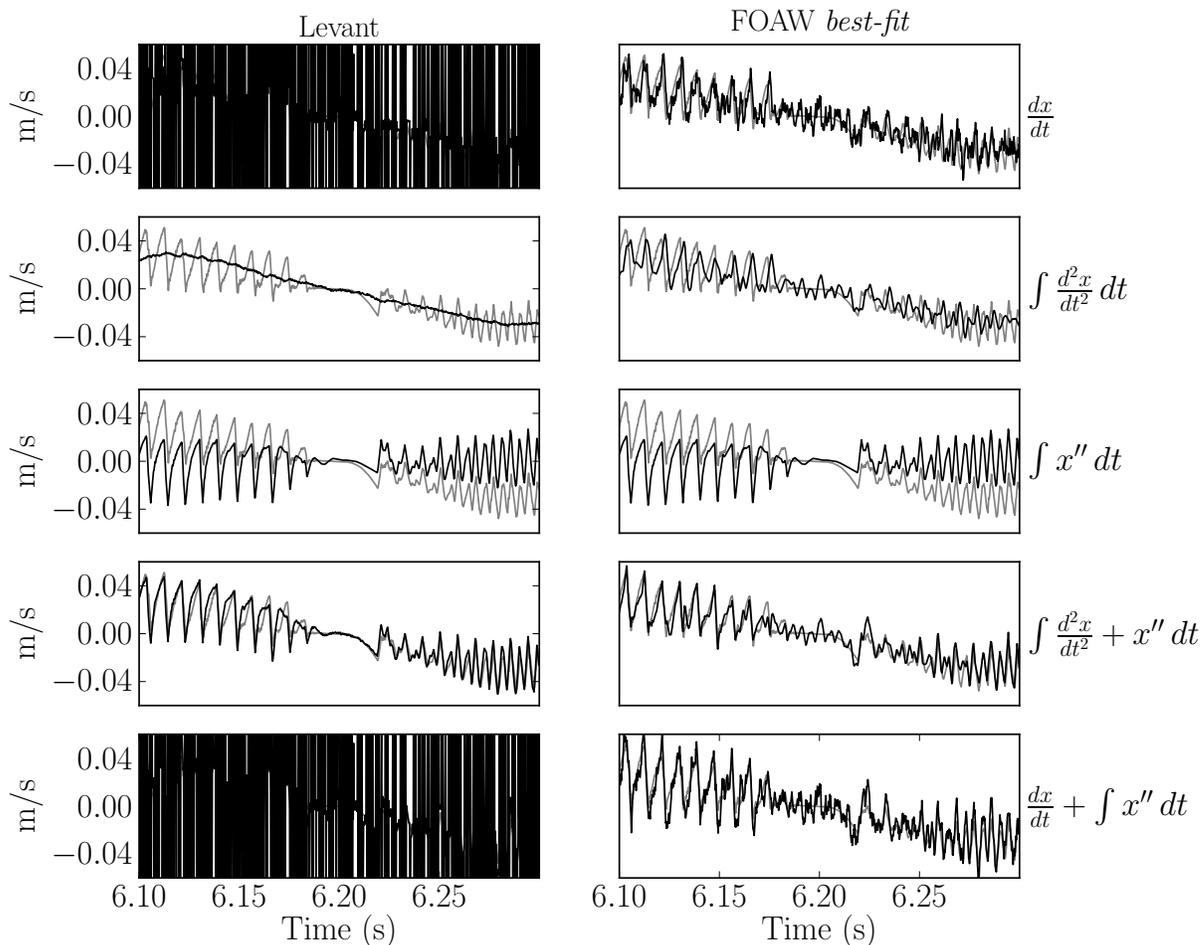


Fig. 5.4 Demonstrating the usage of non-linear differentiators to help correct accelerometer bias. These figures show velocity estimates derived from the integral of double-differentiated acceleration estimates added to the accelerometer measurement. In the labels, x is the position measurement, and x'' is the accelerometer measurement. Left: $\frac{dx}{dt}$ represents Levant's differentiator, $C=100$; $\frac{d^2x}{dt^2}$ represents the differentiator applied twice. Right: $\frac{dx}{dt}$ represents FOAW best-fit, $N = 12$. Comparing the fourth and fifth rows, the difference between single- and double-differentiation + integration is clear in the case of Levant's differentiator, and in the FOAW case there is a subtle improvement in the visible noise. In both cases there is an improvement over the output of single-differentiation (top row) or the pure integration of the accelerometer (third row).

5.5.1 Computational requirements

We have as yet mentioned little about processing requirements. Speaking from an idealised point of view, the computational requirements can be considered unimportant, since all algorithms discussed in this chapter have bounded time determinism, and therefore we can consider it simple enough to choose sufficient computational hardware to execute the preferred algorithm. For example, a routine can be encoded as a digital circuit, e.g. by use of an FPGA or ASIC in the sensor interface circuitry, or performed by a dedicated DSP ahead of the CPU in the pipeline, and executed efficiently without using processor clock cycles needed for simulation. Ideally such a circuit would be built into a commercial haptic device, and the hardware would deliver preprocessed position, velocity, and acceleration data to the user algorithm.

However, in experimental work it is more convenient to use a programmable DSP or CPU, and in this case processing requirements may end up dictating sample rate, or, conversely, sampling requirements may force the choice of algorithm. Therefore, some brief comparison of computational requirements is warranted.

Some of the estimation techniques discussed can be expressed as sets of either FIR or IIR linear filter coefficients. These are the complementary filter, and the least squares FIR fit. An advantage of this is that DSP hardware is often designed to execute such filters especially efficiently. Additionally, executing a kernel filter is cache-friendly and branchless, making it an agreeable choice for modern computational hardware. The least squares fit may require a large number of coefficients, however, depending on the quality desired.

The FOAW algorithm, Levant's differentiator, and the Kalman filter, however, are non-linear routines which require several logical steps per iteration. FOAW *end-fit* requires calculating a two-point linear fit and checking it against the acceptable error margin, once for each point in the maximum window size. Since each check iterates over the window, the worst-case complexity, typically during low acceleration, is $O(N^2)$ in the size of the window. The *best-fit* variant additionally requires performing a least squares line fit, instead of a simple two-point fit, which effectively doubles the number of operations in the main loop. Although typically the window size is not large, we nonetheless found this to be quite prohibitive on our hardware for sample rates higher than 15 kHz or so.

The Kalman filter requires execution of several matrix operations, including the solution of an inverse relationship. This may be computationally demanding, but is an $O(1)$ routine,

since no loops are involved. The largest matrix size for our 1-D, 3rd-order system is 3×3 , so the steps are not exceptionally demanding, and it is fortunately possible to formulate an analytical solution to the inverse equality needed to calculate the Kalman gain, making it unnecessary to use iterative routines such as conjugate gradient. Our C implementation used 198 multiply and 142 add instructions, so it is comparable to a rather large filter kernel. In some cases a Kalman filter can be reduced to an IIR representation, however we made no such attempt, as there is no promise of computational savings from this transformation on our platform.

Levant's differentiator is comparatively simple, consisting only of two numerical integrations per step. In this work we used 4th-order Runge-Kutta integration (RK4), requiring 8 derivative calculations per step, but this is not the only possible choice—a 2nd-order integrator would require half the steps and actually perform similarly, since the switching control does not afford much opportunity for smoother integration. Even with RK4 integration, the number of calculations required by Levant's differentiator is very small compared to the other algorithms we tested.

One caveat is that the second 2-sliding observer state derivative calculation requires the solution of a square root, see sec. 5.3.4. In our work we have used a 100-piece 2nd-order piecewise polynomial fit to approximate the square root in the range of interest, which is in practise much faster than executing the default `sqrt` function in the C math library. This approximation may have led to a minor amount of extra error in the Levant results, however we believe it not to be significant.

Although individually we were able to run these algorithms between 15 kHz to 40 kHz, for testing purposes we needed to run several at once so that switching between them could be done efficiently during subject testing (chap. 7). Therefore we were limited by the hardware; the maximum rate at which we could execute all algorithms together was 5 kHz, and even at this rate, we were forced to switch between each Kalman condition and the FIR filters. The impact was that a subject would occasionally have to wait a second or so for the Kalman state to re-converge when switching to a new condition.

5.5.2 A priori knowledge

Another factor that may influence the choice of algorithm is how each differs in terms of a priori knowledge required about the system conditions, and how elegantly they fail when

incorrectly calibrated.

For instance, the Kalman filter and FOAW require estimates of the noise margin of measured signals. If too high or too low, the Kalman filter tends to drift from the correct value, whereas the FOAW will typically fail to remove noise, leading to a loss in precision, but will not provide inaccurate readings. Fortunately the noise margin can readily be evaluated by measurement, and generally does not change during operation, modulo any electrical or mechanical problems that may develop.

The Kalman filter also requires a process model. We have used a very simple model of a double integrator, which is a common model for a haptic device. More complex models that include friction, damping, or even kinematic information may provide improvements. With some care, closed loop response outside the range of human control may even be taken into account. However, introducing more complex models also means identifying more model parameters a priori. Richard and Cutkosky [104] describe the use of least squares fitting for device friction identification. Kuchenbecker [69] describes methodology for identifying non-linearities in several stages of force feedback devices. We did not apply these methods in this work.

The Levant observer is unique in requiring an estimate of the maximum acceleration expected in the signal. When this estimate is too high, the signal may be inundated with high gain noise; when too low, the signal may temporarily drift quite far from the correct value in response to a sudden transient, before returning.

This acceleration estimate is more difficult to determine a priori as compared to noise margin, since for instance an unexpected acceleration may be experienced if the device is bumped by an external object. Although [28] considers the acceleration expected during normal haptic interaction, in reality this is dependent on the actual feedback algorithm in use, which is subject to change. To ensure robustness, the worst-case acceleration resulting from knocking against the device's physical stops at workspace boundaries should be considered, which will typically be higher than what is expected from a virtual wall algorithm. Otherwise, the tendency to overshoot could lead to high forces and potential instability.

5.6 Conclusion

It is clear that there exist many possibilities for velocity estimation and measurement, many of which have already been proposed for haptic interaction. Some methods are specific to

fixed-position sensing, however even discarding these techniques there are clearly too many options to consider here. We have settled for a selection of examples from particular categories of estimators: polynomial fitting, adaptive filtering, an observer approach, and a stochastic approach. Considering sensor fusion, it may even make sense to use some of these estimators in combination, leading to an explosion of options; we have included one such combination, described in the next chapter.

These choices are mostly based on these methods having been previously proposed for similar applications, and therefore a framework for comparison was needed. Although each method only has 1, 2, or 3 parameters, this is nonetheless quite a large design space, and therefore we developed an optimisation approach based on our criteria of low delay and low noise. If a “best” parameter set can be found for each estimator, only then can we compare the relative performance of each method.

The next chapter performs this optimisation, and Chapter 7 performs an on-line subject-based evaluation of the resulting parameter sets for validation.

Chapter 6

Optimisation and Numerical Evaluation

In order to establish which sensing or filtering technique may provide the most effective solution for high-quality velocity estimation, it was necessary to compare the performance of each proposed solution in simulation based on data recordings. Prior to this, however, we validated that the method for measuring estimation error emphasizes signal qualities identified as important for the target task.

Such an established metric can be used both for automatic tuning of parameters within a given solution, as well as for comparing relative performance between solutions.

6.1 Signal recordings

The TGR was mounted in a 1-DOF horizontal orientation as described in Chapter 3, and configured with the Trans-Tek LVT tachometer and the PCB Piezotronics accelerometer, which were described in Chapter 5. Using the bowed string model, described in 5.1, driven by velocity sampled from the tachometer, 15 recordings of bowing gestures were made at increasing levels of friction force gain.

Each recording is 7 seconds long at 5 kHz, and a selection of these can be seen in Fig. 6.1. All signals were sampled simultaneously to memory by the Toro-16 DSP card at 16-bit resolution and then transferred to host memory via the PCI bus. The signals were then gain-adjusted to match physical units.

These recordings will be used in the next sections as a basis for evaluating the error

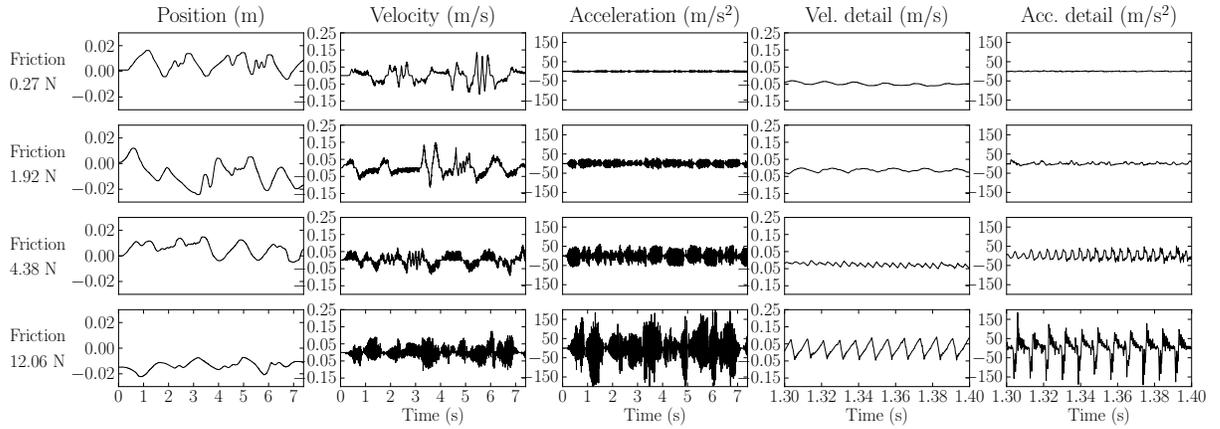


Fig. 6.1 A sample of the recorded signals captured simultaneously with three sensors. Full recording is shown in the first three columns, and zoomed-in details of a section where all recordings are in the steady-state of a bow stroke are given in the last two columns, demonstrating the effect of friction feedback on velocity and acceleration. Both feature increasing amplitude of impulses in response to stick-slip motion of the bow-string friction.

metric, and then to evaluate the velocity estimation methods described in the previous chapter.

Note that we will talk about friction gain in reference to F_{\max} , the maximum available friction force. In the case of the bowed string model, this refers to the height of the rosin curve in the relation $F = \mu R(v_{\Delta}) \cdot v_{\Delta}$, where R is the bow impedance controlled by velocity difference v_{Δ} between bow and string. Thus F_{\max} implicitly determines a gain μ .

The recordings spanned values of F_{\max} from 0 to 12 N. In general, F_{\max} for the bowed string model were higher than values reported in Chapter 3, which went up to about 3 N, because the equivalent of the viscous delimiter (called v_{visc} in Chapter 3) was 0.062 m/s, several times larger than we used for friction testing in that work, thus the slope was shallower.

As mentioned in 5.1, this choice of slope follows from the shape of the transmission coefficient table $\rho(v_{\Delta})$ present in the STK software. We used a constant vertical bow pressure value of $p = 0.1$ throughout these recordings and during testing in the next chapter, which is simply a ratio of the pressure range supported by the STK bow table implementation.

6.1.1 Sensor noise characterisation

One estimation technique that was investigated here, namely the Kalman filter, section 5.4.2, is dependent on assumptions of a Gaussian normal error distribution in the measurement input signal.

This assumption was expected to hold since our apparatus used analog sensors, and the quantization hardware, described in Appendix A, operated below the level of analog noise present in the sampled signal. We also ensured to stay within the linear range of all sensors while recording our example data.

Nonetheless, in order to verify this assumption, we examined the characteristics of our measurements to ensure this approach was justified. We wished explicitly in this work to avoid the need to explore more complex methods that are robust to non-normal input, although in real conditions, e.g. for multi-DOF devices or while using digital sensing hardware, it is possible that methods such as the extended Kalman filter (EKF) may be necessary to account for non-linear distortion in the measurement [14].¹

With the amplifiers turned on and the bowed string algorithm running, the device handle was held still for a few seconds as data was recorded. The resulting distributions can be seen in Fig. 6.2, giving evidence that our assumptions of a normal distribution are justified. We note that the LVT measurement of velocity was not used as input to the Kalman filter in this work.

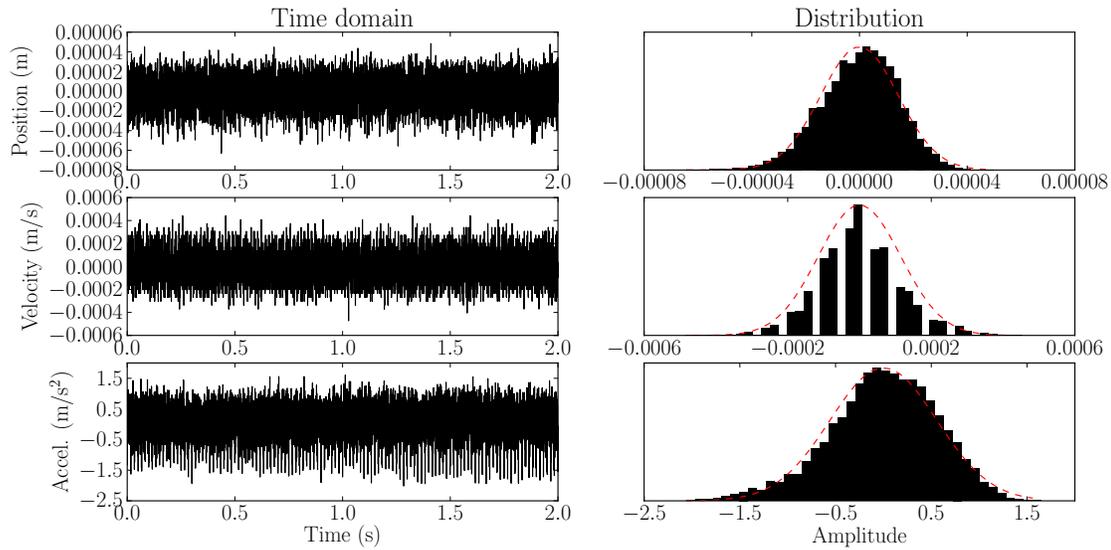
6.2 An error metric for signal comparison

Many works (e.g. [17, 27, 28, 65]) make use of root mean square error (RMSE) to evaluate signal differences. This is calculated for a digital signal as,

$$\text{RMSE}(x_1, x_2) = \sqrt{\frac{\sum_{i=1}^n [x_1(i) - x_2(i)]^2}{n}}. \quad (6.1)$$

In practice we found that using RMSE to compare the performance of velocity estimators often produced a lower error for noisier estimators when compared to estimators

¹The reason for avoiding more complex methods is due to the proliferation of options beyond the simple Kalman filter. An investigation into extended methods could continue almost indefinitely, while in this work we were primarily interested in evaluating solutions that have been proposed in the haptics literature. Further, more robust solutions may be application-specific.

**Fig. 6.2**

A histogram analysis of the noise distribution under the experimental conditions. The device motor amplifiers were turned on and the algorithm was executed at 6 kHz while the handle of the device was held still for two seconds. The recordings of this stationary noise for 3 sensors were detrended, and a 40-point distribution histogram for each is given here, overlaid with a Gaussian fit (dashed line) based on calculations of mean and standard deviation. It can be seen that the match for position and acceleration is quite good, with only a very slight skew towards positive values. The velocity distribution is heavily quantized due to a particularly good signal-to-noise ratio, however the underlying distribution is also clearly Gaussian—nonetheless, the velocity measurement was not used in contexts assuming normal distribution in this work.

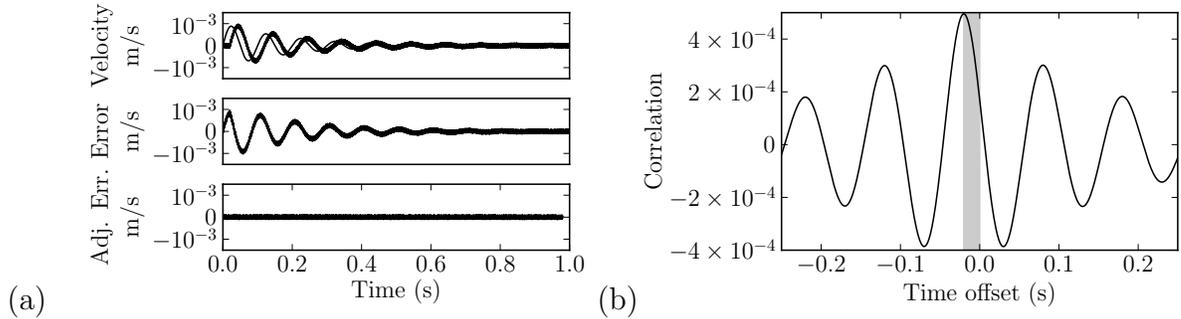


Fig. 6.3 (a) Top: A decaying sinusoid and same signal delayed by 0.02 s and with zero-centered noise of amplitude 2×10^{-4} m/s (RMS= 5.8×10^{-5}) added. Middle: Difference of top signals. Bottom: Difference of original signal and time-adjusted noisy signal. (b) Cross-correlation between the two signals from (a). The indicated offset of the peak correlation from zero is highlighted. For the middle graph of (a), the RMSE is estimated as 2.6×10^{-4} , while for the bottom graph, it is correctly reported as 5.8×10^{-5} .

which imposed some delay. This can be understood intuitively by noticing that RMSE is a summary of the distance between two time-varying signals on the vertical axis. Two signals that are identical but for some delay may show considerable vertical distance during transient moments. In many cases the space created between transient areas due to delay may exceed the distance between a slow-varying curve compared with added noise.

This is demonstrated with a decaying sinusoid in Fig. 6.3(a). It can be seen that the direct difference between the two signals has a height comparable to the original signal—the height of the difference signal in fact depends on the phase difference introduced to the sinusoid by the delay. The time-shifted subtraction removes this phase difference, returning a minimum-amplitude error signal whose amplitude represents our desired metric.

In a virtual wall scenario, we wish foremost to minimize delay, and therefore this may be an adequate comparison metric. However, as we have seen in Chapter 3, depending on the friction level some amount of delay may be undetectable during friction interaction, while very small noise levels may be noticeable. Therefore bare RMSE is not an acceptable error metric for our task, since it is prone to select noisier estimators.

To fix this, the approach we have taken is to independently measure the delay, and use this delay estimate to perform a delay-corrected subtraction, measuring the delay-free error between the two signals. This is accomplished by analysing the cross-correlation between the two signals and finding the peak correlation offset to determine the delay. This delay

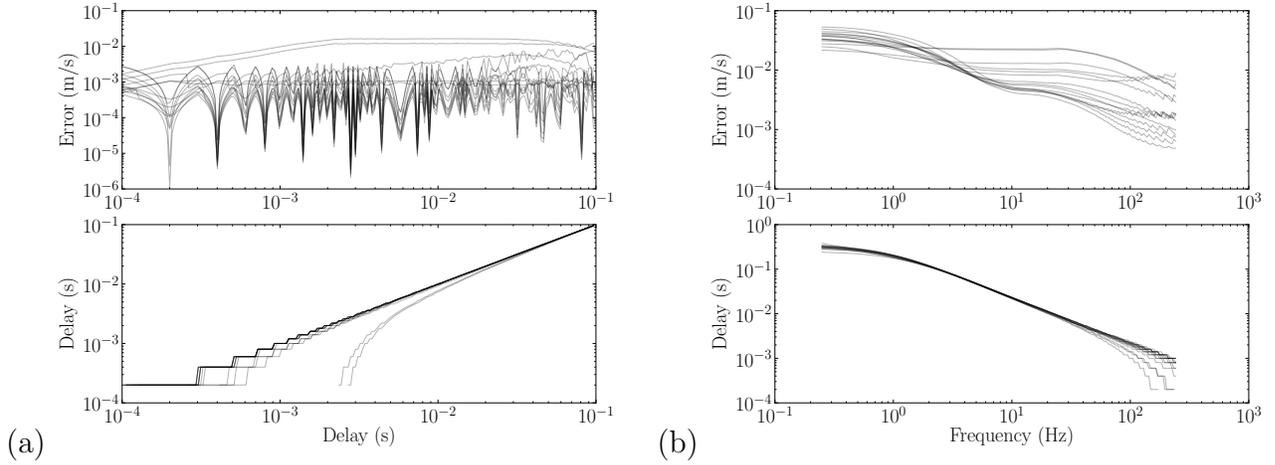


Fig. 6.4 Delay detection and subsequent RMS error evaluation applied to each recording for: (a) pure-delayed versions of the LVT signal; (b) low-pass filtered versions of the LVT signal for various cut-off frequencies. The variance in the error measurement for pure delay, top of (a), is an artifact of sub-sample time shifting used to generate the test signal. Note that there are differences between recordings in how much error or delay is reported for equivalent transformations, particularly for small delay—this is addressed in our multi-objective optimisation approach.

is then removed by time shifting, and the RMSE between the base signal and the delay-corrected signal is measured. These two measures can then be used for optimisation by considering the impact of parameter adjustment on each individually.

6.2.1 Error metric validation

Although Fig. 6.3 shows promising results, prior to using the modified error metric it is necessary to determine whether it will perform adequately on our real data.

Using this method, we measured the delay and resulting error amplitude after compensating delay for each gesture recording against (a) a pure-delayed version of the LVT signal, (b) a low-pass filtered version of the LVT signal. In all cases, the error and delay were estimated for 200 values of delay or filter cut-off frequency, at logarithmically-spaced intervals. For (a), sub-sample delay was implemented via linear interpolation. For (b), a 2nd-order Butterworth low-pass filter was used.

The results, Fig. 6.4 show several positive features. Firstly, in the pure-delayed test (a),

the delay is correctly detected for all signals. The results are seen as quantized at low delay because we do not detect or correct for sub-sample delay.² The error amplitude is also fair since, despite some variance, all signals display an approximately constant result. The reason for the perturbations in the error in Fig. 6.4(a) are due to distortions introduced by linear interpolation, an artifact of the test conditions—this is confirmed by the fact that the same error is introduced for all signals, that the notches in the error occur at sample boundaries, and that similar notches do not appear for the filtered results in (b). Moreover, when we used only sample-aligned delays, error was exactly zero, which we do not show graphically.

In the low-pass filter graph (b), both error and delay decrease as the filter opens up. This corresponds well with the expected characteristics of a low-pass filter. Therefore, we conclude that our error measurement correctly detects and removes delay, allowing comparison of signal distortion amplitude independently from delay.

We note that the strong differences in error seen in (b) demonstrate that the error detection is strongly influenced by particularities of the exact recording being measured. This implies that it is possible to directly consider error differences across estimators within a recording, but comparison between recordings requires normalization.

6.2.2 Comparing velocity signals

Using the error metric determined above, we can compare the results of several parameter sets for the velocity estimators, just as we examined the error and delay results for different cut-off frequencies of a low-pass filter in Fig. 6.4. The goal is to minimize the error in the resulting velocity signal while avoiding the introduction of significant delay.

Minimization requires an error function producing a scalar value. Since our error metric is able to determine delay as well as delay-corrected error, we can attempt to minimize both if we combine them into a single value representing our desire to reduce both simultaneously. This was accomplished by means of a multi-objective optimisation strategy, described in the next section.

²This would be desirable, but errors introduced by sub-sample delay correction by means of linear or cubic interpolation are on the same order as the error we are trying to detect, making the use of sub-sampling time shift compensation infeasible.

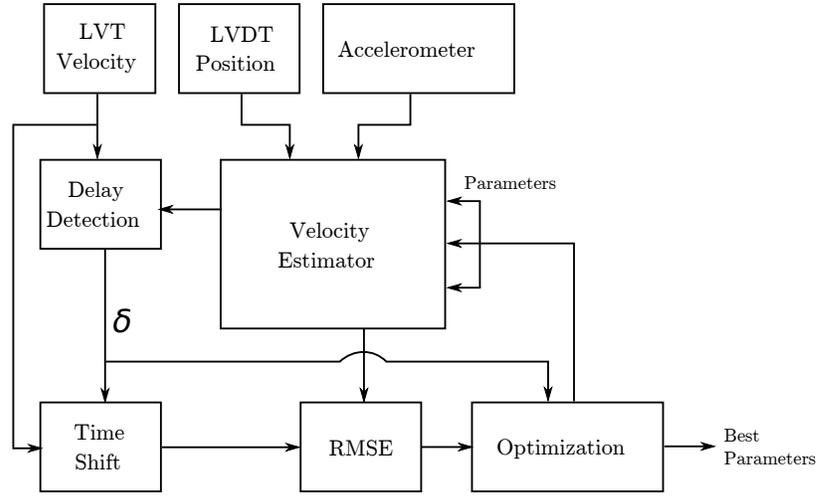


Fig. 6.5 Error evaluation and parameter optimization process.

6.3 Optimization

In this section, we will describe the optimisation strategy used to tune parameters for each estimator. We will compare the resulting error and delay metrics in the following section.

6.3.1 Error surface calculation details

The error function to be minimized is defined as,

$$E(e, \mathbf{x}, r) = \text{RMSE}(v_r, \text{SHIFT}[\tilde{v}_r, D(v_r, \tilde{v}_r)]) \quad (6.2)$$

$$= \sqrt{\frac{1}{n} \sum_k^n (v_r[k\tau] - \tilde{v}_r[k\tau + D(v_r, \tilde{v}_r)])^2}, \quad (6.3)$$

$$v_r = \bar{y}'_r \quad (6.4)$$

$$\tilde{v}_r = e(\mathbf{y}, \bar{y}_r, \bar{y}''_r) \quad (6.5)$$

where D is the relative global delay between two signals detected via a maximum correlation analysis, τ is the sample period, and $v_i[t]$ is time-quantized access into velocity signal v_i at time step $\lfloor t/\tau \rfloor$. We consider the reference v_r equal to the measured LVT signal for recording r , and \tilde{v}_r to be the estimated velocity signal determined by some estimator (e, \mathbf{x}) applied

to position and acceleration measurements, where e is the estimator algorithm and \mathbf{x} is the argument vector to the estimator. Sensor measurements for position (LVDT), velocity (LVT), and acceleration (accelerometer) for recording r are denoted $\bar{y}_r, \bar{y}'_r, \bar{y}''_r$, respectively.

It is important to note that the noise level present in v_r will provide an absolute lower bound on E , since subtracting a noise-free \tilde{v}_r will result in the error level of v_r alone. Thus, the quality of our results is bounded by the quality of the reference LVT recording. We will revisit this in the discussion of the results.

A block diagram describing signal flow during the error evaluation and parameter optimization process can be found in Fig. 6.5.

6.3.2 Multi-objective global optimisation

Challenges in optimising the parameters of each estimator were two-fold: firstly, since in several cases the error surfaces were not strictly convex and also noisy, it was not possible to use a simple gradient-descent optimisation strategy since local minima was an issue; secondly, we wished to determine the best parameters to minimize both error and delay, which typically are trade-offs and therefore difficult to consider simultaneously.

Therefore, we chose to combine a stochastic global optimisation approach with an *objective sum* strategy for multi-objective optimisation. This allowed to generate a possible best parameter set for each estimator, and also serves as a tool for comparing their performance numerically. Marler and Arora [77] provide a useful survey of multi-objective methods, and we briefly summarize their notation in the following paragraphs in order to contextualize this choice.

In multi-objective optimisation, a set of objective functions $F_i(\mathbf{x})$ must be simultaneously minimized, where $\mathbf{x} \in E^n$ is a parameter set in the space of *design variables*, where n is the number of variables and E is a Euclidean space. Usually this is accomplished by combining them into a single objective vector $\mathbf{F}(\mathbf{x}) = [F_1(\mathbf{x}) \dots F_k(\mathbf{x})]^T$, and reducing this by some means to a scalar function $F_g(\mathbf{F})$, known as the *global objective function*, which can be minimized.

Importantly, we must consider that an optimisation strategy is not guaranteed to give a uniquely “best” solution—multiple solutions may be available. This can be understood in the context of *Pareto optimality* [93]: a point $\mathbf{x}^* \in \mathbf{X}$ is Pareto optimal if and only if there are no other optimal points in \mathbf{X} , where $\mathbf{X} \subseteq E^n$ is the *feasible design space*. Often,

this point does not exist; rather, we must consider *weakly Pareto-optimal* points. A point \mathbf{x}^* is weakly Pareto-optimal if there exists no other point $\mathbf{x} \in \mathbf{X}$ such that $\mathbf{F}(\mathbf{x}) < \mathbf{F}(\mathbf{x}^*)$. Finally, a point $\mathbf{x}^* \in \mathbf{X}$ is *Pareto efficient* if and only if there does not exist another point $\mathbf{x} \in \mathbf{X}$ such that $\mathbf{F}(\mathbf{x}) \leq \mathbf{F}(\mathbf{x}^*)$ with at least one $F_i(\mathbf{x}) < F_i(\mathbf{x}^*)$.

In other words, Pareto efficiency identifies the conditions in which no one objective can be improved without sacrificing another objective. The distinction from weakly Pareto optimal is that the latter identifies points for which no alternatives exist where *all* objectives can be improved.

In a multi-objective optimisation, the set of efficient points, called the *Pareto-efficient frontier* is what we seek to determine. Some characteristics of the global criterion may determine was subset of the Pareto-frontier is discoverable.

Global criterion selection

We note the above because our selection of F_g affects our ability to completely determine the full efficient frontier. Specifically, for reasons of simplicity, we have chosen to use the standard objective sum method for combining our objective functions. This is a special case of the *weighted sum* method [77],

$$F_g(\mathbf{x}) = \sum_{i=1}^k \frac{w_i F_i(\mathbf{x})}{s_i} \quad (6.6)$$

where all weights $w_i = 1$, and s_i is a normalization scaling factor. Other choices of weights would allow expression of preference for some objectives over others.

It should be noted that although this method can identify a weak Pareto optimum, it does not guarantee that it is unique; moreover, it can be shown that this method will not produce, by weight adjustment, an evenly-distributed representation of the Pareto-efficient set [77]. Nonetheless there exist methods for adapting weights dynamically to improve Pareto frontier generation [67].

Therefore, we selected this method with the assurance that some Pareto-efficient point will be selected for each run of our optimiser, but with the caveat that there may be some external criteria, for example the processing requirements or implementation complexity, which could be additionally considered and for which a larger set of Pareto-efficient points would be interesting to see. These criteria could of course be included as additional objec-

tive functions, if they can be quantified and weighted appropriately.

Nonetheless, this remains a caveat of the multi-objective optimisation procedure: that we are determining the best possible set of parameters for each estimator according to our two-dimensional criteria of error and delay does not imply that we will find the only choices available. Thus, although in this work we will not further-tune the output of the optimiser, there may be other criteria and other multi-objective methods to consider that could yield better choices. This will be addressed further in our discussion at the end of this chapter.

Normalization

In order for $w_i = 1$ to be effectively true, the objective functions must be of comparable magnitude. In this work, the objective functions comprise the amount of error and delay determined for each recording. Not only are error and delay of different units, but between recordings we can expect the estimates of these values to vary somewhat; in particular, we expect stronger friction forces to be more sensitive to distortion in terms of detected error. More generally, higher, rapidly-changing velocities will result in a different signal-to-noise ratio profile than for steadier, smoother signals.

Therefore each objective must be independently normalized by setting s_i appropriately, so that they can be summed without bias. This can be done by estimating in advance the lower and upper bounds expected for each F_i .

However, in our case some parameter sets may result in divergence, particularly for observer-based methods, meaning our upper limit is not reliable, and our lower bound is unknown. On the other hand, since we are simultaneously evaluating several estimation methods, and are interested principally in how they compare with each other, we found it convenient to dynamically set the scaling and offset according to the best- and worst-performing estimators.

Therefore, on each iteration k , we compute $E_{e,r}$ and $D_{e,r}$, the error and delay for

estimator e for recording r , and compute,

$$M_{E,r}(k) = \max_{0 < j \leq n} E_{j,r}(\mathbf{x}_e(k)) \quad (6.7)$$

$$m_{E,r}(k) = \min_{0 < j \leq n} E_{j,r}(\mathbf{x}_e(k)) \quad (6.8)$$

$$M_{D,r}(k) = \max_{0 < j \leq n} D_{j,r}(\mathbf{x}_e(k)) \quad (6.9)$$

$$m_{D,r}(k) = \min_{0 < j \leq n} D_{j,r}(\mathbf{x}_e(k)) \quad (6.10)$$

$$F_{2r}^e(k, \kappa) = \frac{E_{e,r}(\mathbf{x}_e(k)) - m_{E,r}(\kappa)}{M_{E,r}(\kappa) - m_{E,r}(\kappa)} = \frac{E_{e,r}(\mathbf{x}_e(k)) + o_{E,r}(\kappa)}{s_{E,r}(\kappa)} \quad (6.11)$$

$$F_{2r}^e(k, \kappa) = \frac{D_{e,r}(\mathbf{x}_e(k)) - m_{D,r}(\kappa)}{M_{D,r}(\kappa) - m_{D,r}(\kappa)} = \frac{D_{e,r}(\mathbf{x}_e(k)) + o_{D,r}(\kappa)}{s_{D,r}(\kappa)} \quad (6.12)$$

where $(s_{E,e}, s_{D,e}, o_{E,e}, o_{D,e})$ are scalings and offsets for error and delay for each recording across all estimators. This normalizes error and delay functions for n estimators into the range $[0, 1]$ at iteration k , based on the results of some (possibly different) iteration κ . We wish to minimize $F_g^e = \sum_{j=0}^k F_j^e$ for all estimators e , and finally rank the estimators accordingly.

Since our scale changes dynamically, it is of course important to keep the scale the same when comparing across iterations. We proceed then by comparing $F_g^e(k, b_k) < F_g^e(k+1, b_k)$, where b_k is the iteration of the best-so-far F_g^e at iteration k . Stated algorithmically, the steps are as follows:

1. At iteration $k_0 = 0$, for each estimator,
2. Select parameter set $\mathbf{x}_e(k_0)$.
3. Compute $E_{e,r}$ and $D_{e,r}$ for all estimators on all recordings.
4. Let $b_0 = k_0$, and memorize $s_{E,e}(b_0), s_{D,e}(b_0), o_{E,e}(b_0), o_{D,e}(b_0)$.
5. For every subsequent iteration k ,
6. Select parameter set $\mathbf{x}_e(k)$.
7. Compute $E_{e,r}$ and $D_{e,r}$ for all estimators on all recordings.
8. Compare $F_g^e(k, b_k) < F_g^e(k+1, b_k)$.
9. If true, let $b_{k+1} = k$ and memorize the scalings and offsets. Otherwise, let $b_{k+1} = b$.

In this manner, successive comparisons are based on similar scaling, yet the scaling is adjusted as the performance of all estimators globally improves. This is important, since if

we based the scaling statically on the result range of the first iteration only, for example, it would not be representative of the true value range, because the first few iterations may be very different in range from the near-optimised results. Note that in the case of divergence, we replace infinities or undefined results with large numbers.

One way of thinking about this procedure is that at each iteration the estimators are ranked against each other, and one after the other we perturb a parameter set and retain it if there is any improvement. This causes each estimator to compete against the others, trying to bump itself up the ranks as much as possible. As stopping condition, we halt if no improvements have been made to any estimators in 1000 iterations.

Global stochastic optimisation

Finally, we must discuss the method of selecting our parameter sets at each iteration, so as to iteratively close in on an optimal choice. Possibilities here can be divided into some categories: those of gradient-based approximation, and gradient-free; and those of a deterministic and stochastic nature [109]. Further, stochastic approaches can consider a *stochastic model*, e.g. probabilistic sampling of the gradient, or can be based on *stochastic sampling* of points in the domain.

In general, gradient-based approximators require convex objectives to ensure convergence, while gradient-free approaches use some concept of “attraction” to the lowest-found point without calculating a gradient. Stochastic approaches can be used to avoid local minima, and increase the likelihood of finding a global optimum in noisy data, by randomizing the parameter set and smoothing out small perturbations in the error surface. Moreover, for large dimensionality, they are often considered more feasible than deterministic approaches such as brute force grid search [109].

Since local minima was indeed a problem for us, and we could not even guarantee differentiability in some cases, we sought the use of a gradient-free approach. We avoided the use of complex methods such as swarms, genetic algorithms, and simulated annealing [137]. Instead, although it was not the most efficient possibility in terms of total evaluations of the objective function, we used a very simple method of *pure adaptive search* [95], which is a model-free stochastic sampling approach. At each iteration a point within a bound is randomly sampled, and over several iterations this bound is reduced by some amount, centered around the best-so-far discovered point each time. Thus a successive

approximation based on random sampling of nested regions is used to zoom in on the optimal choice, sampling the region around it with increasing density.

Although for large dimensionality this would be inefficient to the point of intractability, since we were working with only 1 and 2 parameters at most, we decided to use this method because it ensures a thorough, unbiased sampling of the objective function without the need for much tuning of the optimiser.

We reiterate the fact that the optimiser alone is not guaranteed to give the best general result, since the problem of determining good arguments to F_i is in fact ill-posed, in the sense that very similar objective functions, by some measure of distance, can give quite different estimates for the arguments [109]. During this work, we witnessed this by the fact that some of our objective functions have fairly flat valleys, such that small differences in the optimum resulted in drastic changes in the resulting parameters; others featured global optima which were unique to one or a small subset, and therefore not generalizable. (See Appendix B for an overview of these issues.)

Indeed, this is related to the problem of over-fitting, and therefore we show that this problem is overcome by the use of multi-objective methods, which try to optimise several objectives and thereby “average out” too-specific cases for any one recording. We also verify this by checking our results against a separate data set.

6.4 Results

Results of the optimisation give a parameter set for each estimator, from which we can calculate a global ranking as well as evaluate individual performance for each recording.

In this section, we first make some remarks regarding the selected parameters, and secondly we provide an analysis of error and delay performance under the optimised conditions.

Table 6.1 provides a summary of the estimators tested, which were described in detail in Chapter 5. These labels will be used throughout the remainder of this chapter for discussion of optimisation results.

6.4.1 Parameter selection

Final parameter selection based on the global criterion is given in Table 6.2. There are several questions we can ask regarding the selected parameters:

<i>Estimator</i>	<i>Description</i>	<i>Parameters</i>
COMP1	Sum of a 2 nd -order Butterworth low- and high-pass complementary filter pair, configured with identical cut-off frequencies	LPF=HPF
COMP2	Sum of a 2 nd -order Butterworth low- and high-pass complementary filter pair, with independently-optimised cut-off frequencies.	LPF, HPF
FOAW	First-order adaptive windowing filter, <i>best-fit</i> method.	Max. window size, noise margin
KALPOS	2 nd -order Kalman filter with a double-integrator process model, position measurement input, position covariance set to $R_p = 2 \times 10^{-10}$.	Process covariance Q .
KALMAN	3 rd -order Kalman filter with a double-integrator process model, position and acceleration measurement input, position covariance set constant at $R_p = 2 \times 10^{-10}$.	Process covariance Q , accel. covariance R_a .
KALLEV	Similar to KALMAN, with acceleration measurement replaced with double-differentiated position using LEVANT.	Process covariance Q , accel. covariance R_a .
KALLEVACC	Similar to KALMAN, with acceleration measurement added to double-differentiated position using LEVANT, $C = 100$.	Process covariance Q , accel. covariance R_a .
KALFOAW	Similar to KALMAN, with acceleration measurement replaced with double-differentiated position using FOAW of size 12, noise margins 10^{-4} m and 0.5 m/s.	Process covariance Q , accel. covariance R_a .
KALFOAWACC	Similar to KALMAN, with acceleration measurement added to double-differentiated position using FOAW of size 12, noise margins 10^{-4} m and 0.5 m/s.	Process covariance Q , accel. covariance R_a .
LEASTSQ	Least squares polynomial fit expressed as a set of FIR filter coefficients.	Poly. order N , window size M .
LEVANT	Levant's differentiator, a 2-sliding observer driven to follow an input position signal.	Max. accel. C
LEVANTLP	The LEVANT estimator followed by a 2 nd -order low-pass Butterworth filter.	Max. accel. C , LPF cut-off freq.
LOWPASS	A simple 2 nd -order Butterworth LPF.	LPF cut-off freq.

Table 6.1 Complete list of estimators tested, with description and a list of parameters for each. See Chapter 5 for details.

<i>Estimator</i>	<i>Parameters</i>
COMP1	LPF = 317 Hz HPF = 317 Hz
COMP2	LPF = 256 Hz HPF = 2254 Hz
FOAW	Size = 17 Noise = 1.001×10^{-4} m/s
KALPOS	$Q = 0.191$
KALMAN	$Q = 108$ $R_a = 6.28 \times 10^4$
KALLEV	$Q = 6.38 \times 10^4$ $R_a = 195$
KALLEVACC	$Q = 2954$ $R_a = 64.6$
KALFOAW	$Q = 77.7$ $R_a = 8.06$
KALFOAWACC	$Q = 88.5$ $R_a = 3.85$
LEASTSQ	$N = 2$ $M = 33$
LEVANT	$C = 19.7$
LEVANTLP	$C = 3.6$ LPF = 293 Hz
LOWPASS	LPF = 256 Hz

Table 6.2 Final parameters for each estimator based on global optimisation across all recordings. Note that position covariance $R_p = 2 \times 10^{-10}$ was set a priori for all Kalman filter estimators, based on measured variance. All other parameters were free variables during optimisation. Implicit here is that accelerometer input for Kalman filters was additionally pre-filtered with HPF = 20 Hz to remove bias. Estimators and their parameters are defined in section 6.4.1.

How well do parameters related to physical values or measurement variance match manual tuning?

FOAW For FOAW, the noise threshold reflects the expected amplitude of the position signal noise at low velocity. This is used as an error margin to judge selection of a smallest valid window size for line fitting. Since our position noise was measured to have a maximum error of 6.4×10^{-5} m, (RMS amplitude 1.5×10^{-5} m,) we should expect a value for the FOAW error margin in approximately that order of magnitude. The optimal error margin was selected as 1.001×10^{-4} , which is about 50% larger than expected, however it is well within the expected range.

LEVANT In manual tuning, one would set C , the Lipschitz constant representing a bound on acceleration, to the worst-case expected value during interaction, to ensure robustness. However, since our test recordings consisted of several values for friction force gain, the maximum acceleration is different for each case, see Fig. 6.6. Therefore the optimiser had a difficult job of choosing C since it cannot take a worst-case approach across recordings. Nonetheless, a relatively small value compared to the measured maxima was selected for bare LEVANT, and an even smaller one for LEVANTLP. On inspection, we found that with the selected parameters, the switching noise of the 2-sliding observer could exceed the amplitude of velocity impulses driven by bow-string slip events, and additionally at periods of high acceleration it could indeed exceed the Lipschitz bound, as demonstrated in Fig. 6.7. To reduce the error criterion, the optimiser would select as small a C as possible. Thus, as long as post-filtering can cover these problems it is unlikely to consider this divergence as a significant issue without special consideration.

KALMAN For the Kalman-related filters, it is possible to set the measurement covariance parameters R based on measured noise amplitude. The process covariance Q is more difficult to tune and it is natural to determine it using an automatic optimisation procedure. However, we elected to hold position covariance constant but optimise acceleration covariance, since the DC-biased behaviour of the accelerometer was expected to affect its role in the Kalman prediction model. Note that since these covariance parameters ultimately affect the Kalman gain which sums to 1, it is their *relative* values that affect behaviour of the Kalman update, therefore it is necessary to hold at least one

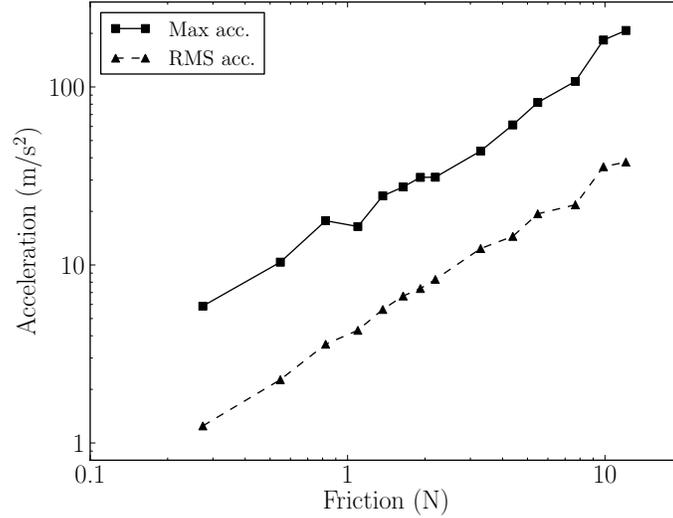


Fig. 6.6 Maximum and RMS acceleration vs. friction for each recording as measured by the accelerometer.

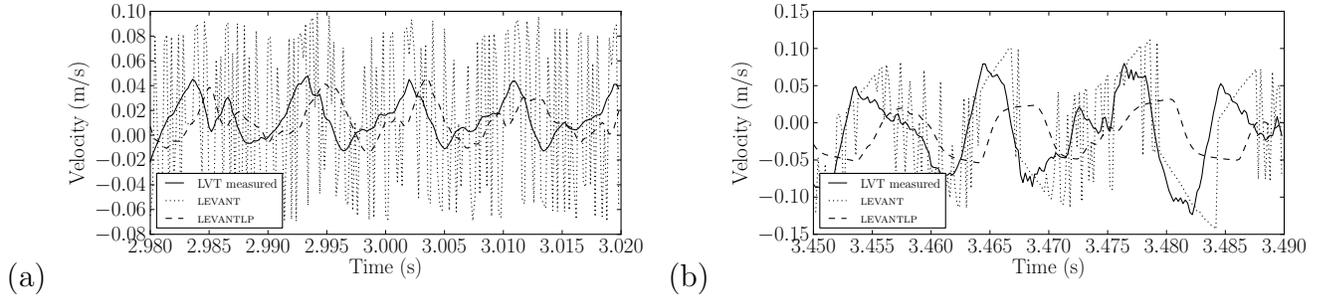


Fig. 6.7 Examples of Levant's differentiator behaviour. For both graphs, $F_{\max} = 7.67$ N, LEVANT $C = 19.7$, LEVANTLP $C = 3.6$, LPF = 293 Hz. (a) Switching noise exceeds slip amplitude. (b) Lipschitz bound is exceeded for short intervals, e.g. from $t = 3.480$ to 3.487 s. In both cases, the low-pass post-filtering effectively removes switching noise and covers up temporary divergence, allowing for a smaller choice of C . The trade-off is an increase in delay and a reduction of the sharpness of the peaks.

parameter constant. See Appendix B for more details. For this reason, position covariance was held constant at $\sigma^2 = 2 \times 10^{-10}$, where $\sigma = 6.4 \times 10^{-5}$ m/s was the measured noise amplitude.

None of the covariance values selected are in the same order of magnitude as the measured variance of $\sigma_a^2 = 0.048$, ($\sigma_a = 0.22$ m/s².) However, within a few orders, we found that the relative values of Q and R_a have a more significant effect than the relation to R_p . Their choices are discussed below.

What can we conclude about quality of sensing from parameter selection?

Examining the relative values of Q and R_a in Table 6.2, we notice that in the simple KALMAN filter where position and acceleration measurements are provided directly, R_a was selected as 2 orders of magnitude higher than Q . This shows great distrust of the acceleration signal, likely due to DC bias.³ On the other hand, when the accelerometer bias was corrected in the hybrid techniques (KALLEVACC, KALFOAWACC) by adding it to a double-differentiated position signal, trust in the acceleration signal is improved, such that the process covariance becomes 1 or 2 orders of magnitude higher than measurement covariance.

More surprising is the apparent lack of trust of the accelerometer exhibited by the complementary filter cut-off frequency. When the same frequency is used for both low- and high-pass filters, estimator COMP1, a fairly high value of 317 Hz was selected, which is in fact higher than the cut-off frequency selected for the simple LOWPASS condition. Additionally, when given the opportunity to choose a separate high-pass filter, estimator COMP2, a very large frequency rejection is selected at 2254 Hz, along with an identical position filter to the LOWPASS estimator. This would seem to indicate that the accelerometer data was not found to be helpful during estimation. This is perhaps confirmed by seeing that the R_a selection for KALFOAW and KALFOAWACC is approximately similar, although this does not hold true between KALLEV and KALLEVACC.

However, before drawing conclusions about the quality of our acceleration measurements, a performance comparison must first be made, which can be found in the next subsection.

³Note that measured DC offset from zero was removed by a high-pass pre-filter at 20 Hz, which may have imposed inaccuracy as compared to a simple subtraction of the mean, however this was necessary to ensure robust integration.

For more investigation into the role of parameters for each estimator, see Appendix B, which provides visualizations of the effect of the parameter space on error and delay performance.

6.4.2 Performance

In order to summarize error and delay performance for each estimator, we first looked at raw results for each recording. This can be found in Fig. 6.8. It is clear from this representation that there are some trends for performance within specific recordings.

Therefore, to consider our data set as comparable so that a summary could be made, these values were first normalized according to our comparison case, LOWPASS. In Fig. 6.9, the same results as Fig. 6.8 are given, expressed as a percentage of the LOWPASS results. It can be seen that there is far more regularity in this representation, with the exception of the highest-frequency recordings and some high values for delay due to division by small values. These small values result from replacing zeroes in the detected delay which occurs because of quantization to sampling instants.

In order to display results in a manner insensitive to these outliers, we used a non-parametric box-and-whisker plot to show results summarized across all recordings, except those with $F_{\max} = 9.9$ and 12.1 N—these added biased variance to the results and are in any case outside the range of interest for bowing simulation, therefore they have been excluded. Final results are seen in Fig. 6.10, showing overall performance of the multi-objective criterion, ordered best (lowest) to worst (highest). In the same order, median and quartiles of error and delay are also shown. Further, while comparing across conditions, it is important to consider significance of differences between pairs, therefore we refer also Fig. 6.11, which shows significance according to a Wilcoxon rank-sum test [91].

From these results it can be seen that the multi-objective optimisation has correctly balanced the two criteria: the top performers on the global criterion also performed best for both error and delay.

With some exceptions, most approaches performed as well as or better than LOWPASS on both error and delay criteria, in some cases with a median improvement of about 30% on error, and 50% on delay.

The 3rd-order Kalman estimators were the best performers on both criteria. Differences between them, however, were generally not significant. The performance of the hybrid

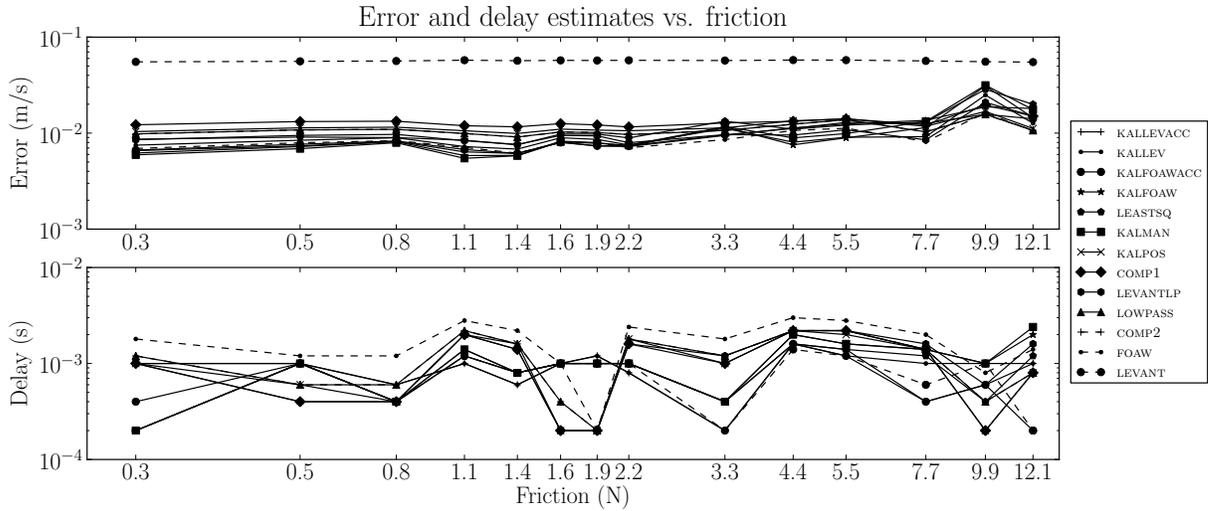


Fig. 6.8 Error and delay evaluation for each recording using optimised parameter sets for each estimator. All estimators, with the exception of LEVANT, follow trends per recording. Therefore, these results are normalized by dividing into LOWPASS results, shown in Fig. 6.9, and are presented more legibly in a summarized form in Fig. 6.10.

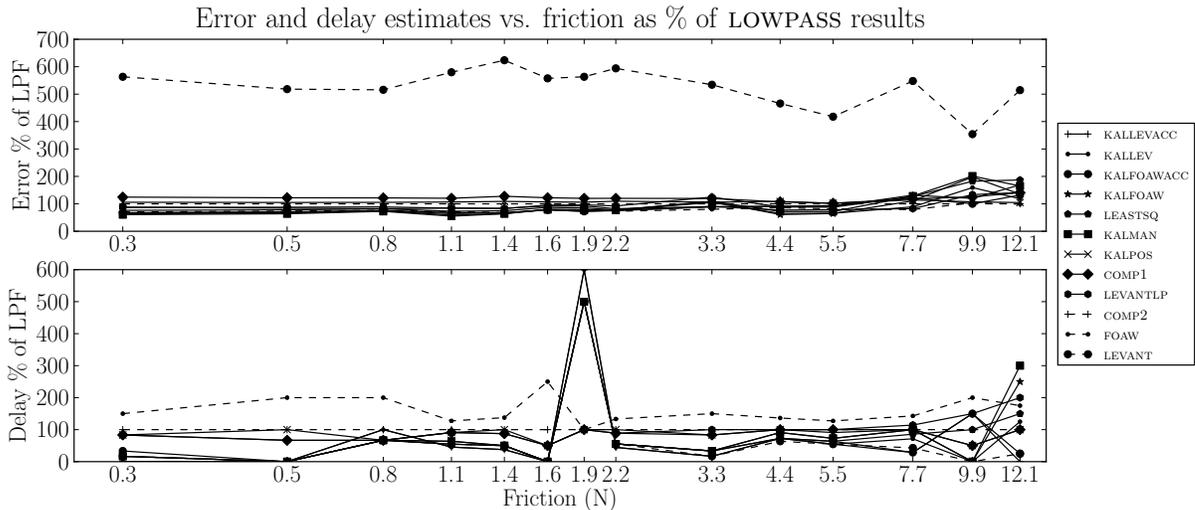


Fig. 6.9 Error and delay evaluation for each recording using optimised parameter sets for each estimator, normalized by dividing into the results of LOWPASS. Some delay results are large due to division by small values that replaced delay detected as zero. To improve readability and to remove these outliers, these data were summarized in non-parametric form in Fig. 6.10.

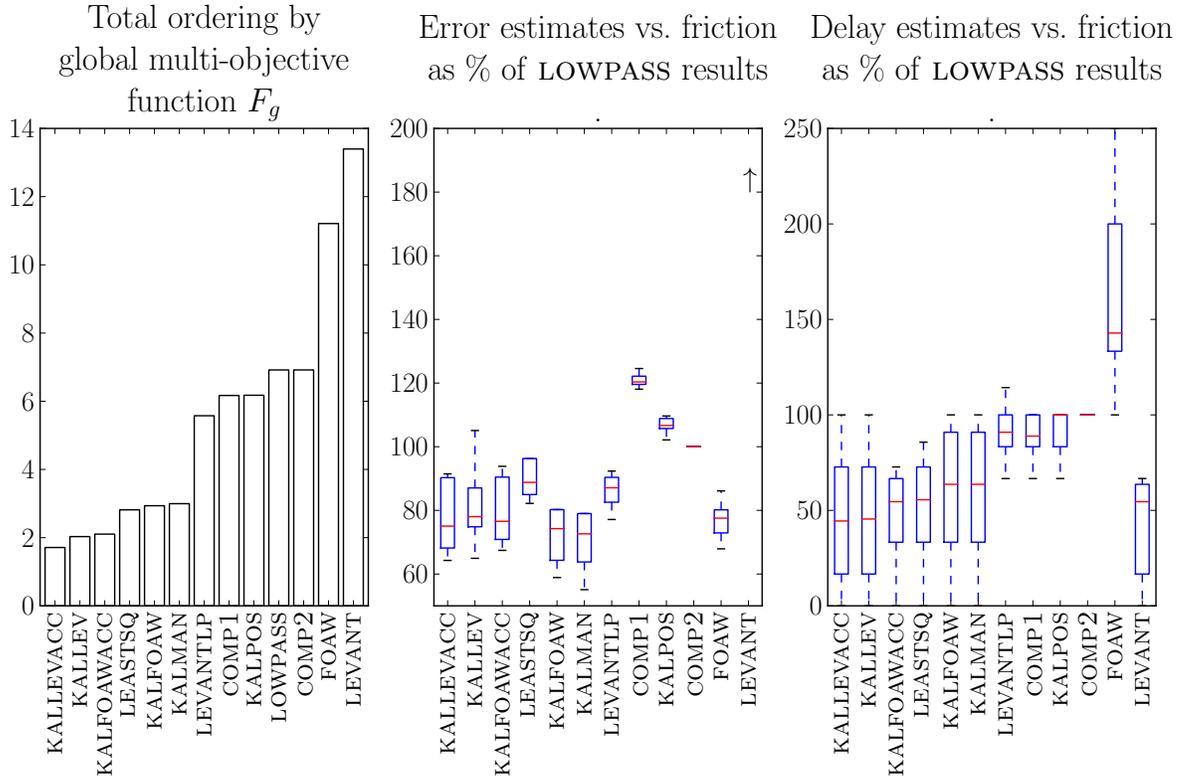


Fig. 6.10 Global objective performance, left, with error and delay evaluation, middle and right, for each recording using optimised parameter sets for each estimator, non-parametric box plot form. Results are given in left-to-right order of best (lowest) to worst (highest). Note that LEVANT error results are present above the graph limits, with a median of 560%.

approaches—the Kalman filter combined with non-linear estimators—indicate some trends toward improvement in delay, but were not better on error. This improvement in delay for the Levant differentiator-driven Kalman filters, KALLEV and KALLEVACC, brought a small margin of improvement in terms of the global criterion, as compared to KALMAN; however, we cannot claim significance of these results due to variance.

This brings us to the evaluation of the accelerometer, which we cannot show to bring an improvement to the velocity estimation. Although the KALMAN estimator performs much better than KALPOS, examination of the Kalman gain reveals that the position measurement receives greater weight. Additionally, there is little difference between the performances of KALLEV vs. KALLEVACC and KALFOAW vs. KALFOAWACC, except a non-significant im-

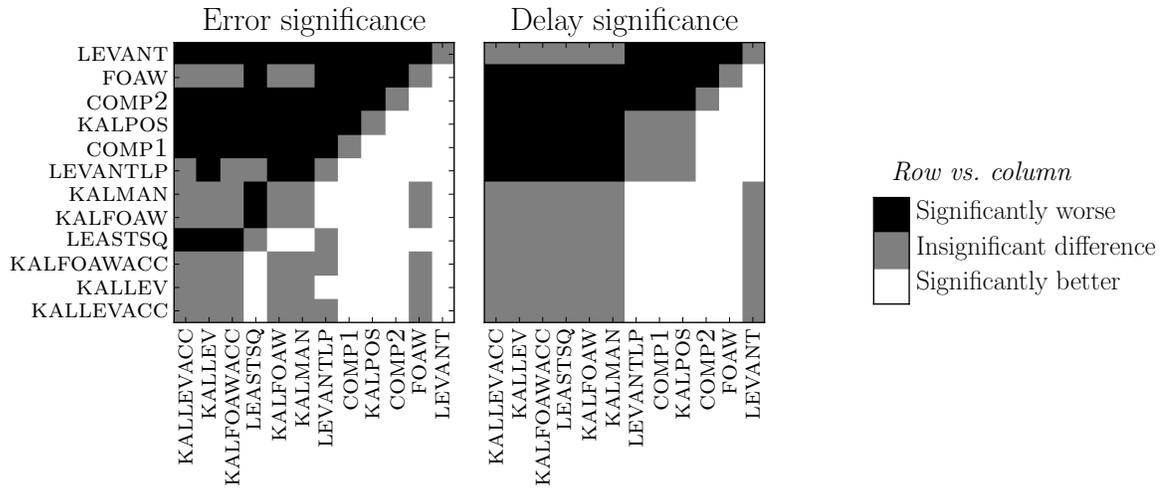


Fig. 6.11 Error and delay evaluation, significance of differences according to Wilcoxon rank-sum test. Significance category was determined by $p < 0.05$, indicating rejection of the null hypothesis that two estimators have the same distribution.

provement in delay in the latter.

More tellingly, the COMP2 estimator performs approximately identically to LOWPASS, while COMP1 experiences a fairly strong increase in error due summation with the accelerometer data. We must conclude that the accelerometer did not provide useful information for velocity estimation in this experiment.

6.4.3 Validation

A separate data set was recorded with the same friction values and very similar gestures as our test set, and kept aside for validation. Using the parameter sets from Table 6.2, we evaluated error and delay for the second data set.

Taking the difference between results at matching friction levels, the average percentage of absolute error difference was 10.3% of the test set results, with a standard deviation of 7.76%. The average percentage of absolute delay difference was 75.6%, with a standard deviation of 102%.

This large deviation in percentage delay between data sets is also reflected in the large deviations between recordings in Fig. 6.10. At delays of only a few samples, variation is somewhat expected since delay values are small and quantized, so differences of one or

two samples can account for large percentage differences. Between the two data sets, no more than 7 samples of delay difference actually occurred, with an average of 2.60 absolute samples across all recordings, and a standard deviation of 2.02 samples. Differences in mean delay between data sets was not significant across estimators, but there was a correlation between the mean delay absolute difference and standard deviation, (Pearson's $r = 0.72$, $p < 0.05$.) Small mean delays received smaller deviation in delay, meaning that time-accurate estimators remained accurate for the validation data set.

Therefore we can conclude that the performance of our estimators on data outside the training set is within a satisfactory margin of error.

6.4.4 Execution time

We benchmarked each estimator algorithm on the parameters in Table 6.2. The time for each estimator to run 1000 times consecutively was measured 3 times for each estimator for each recording, giving a total of 45 data points per estimator. Results were normalized by the number of samples in each recording, in order to give a result in terms of $\mu\text{s}/\text{sample}$. This was performed on an 3.4 GHz Intel Core i7 computer, using only 1 core. Algorithms were written in C and compiled with gcc with optimisation level 2, although execution was managed by a Python program. All algorithms were executed in a per-sample loop in C over a whole recording per run, therefore interpreter overhead should be negligible.

It is clear from the results, shown in Fig. 6.12, that timing was dominated by the estimator run times, therefore these times can be considered a good representation of the execution cost of each algorithm. We can see that in terms of computational requirements, Levant's differentiator performs better than even the complementary filters. The FOAW algorithm has a significant run-time cost since, on our data, it tends to maximize its usage of the window, meaning an exhaustive search through every sub-window on each sample.

6.4.5 Discussion

Two surprises in these results for us were the excellent performance of the LEASTSQ estimator, being effectively comparable with Kalman approaches, and the poor delay performance of the FOAW estimator.

The 2nd-order least squares fit turns out to be a very effective estimator in terms of delay. Although its error performance is not as good as other methods, it proves to have good

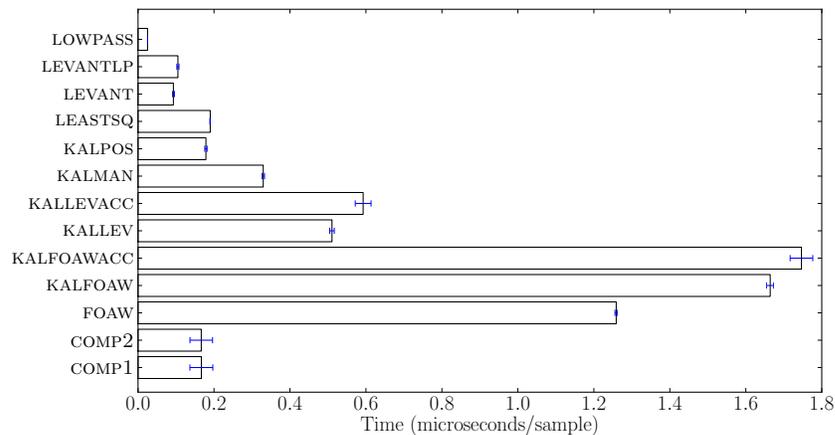


Fig. 6.12 Comparison of execution times for each estimator. It can be seen that the speed of Levant’s differentiator is the most computationally efficient method after a simple *bw2* low-pass IIR filter. Hybrid methods tend to be more computation-hungry, since they apply differentiators twice and then combine them using a Kalman filter.

enough delay qualities that it achieved a very competitive global ranking. In comparison, the FOAW estimator had very good error performance, but suffered badly from delay.

This was surprising because the FOAW algorithm is designed specifically to improve on delay and accuracy during transients—it uses a long window during low velocity moments, but decreases the window size *just in time* as transients are encountered. If we consider that our data consists of continually triggered small transients due to stick-slip action during each stroke, and that no large, single-event impulses are experienced as in virtual wall interaction, it seems likely that a window size of several samples might be selected *on average*. Moreover, except for cases of very strong feedback gain, the shape of the velocity signal is essentially triangular, rather than like a step function, and therefore may not consistently demand sharp decreases in window size at each slip.

As opposed to LEASTSQ, which generally improves the locality of its estimate with window size as the curvature of the 2nd-order fit is improved, FOAW is essentially a line fit, and therefore its estimate corresponds approximately with the center-point of the window, leading to delay if an appropriate window size is not selected. In practice, unless the window size is modulated with the stick-slip action, we can assume that this capability is not being well-utilized to improve performance during a bow stroke. To test whether this was the case, we looked at window size selected by FOAW over our recordings. For all recordings,

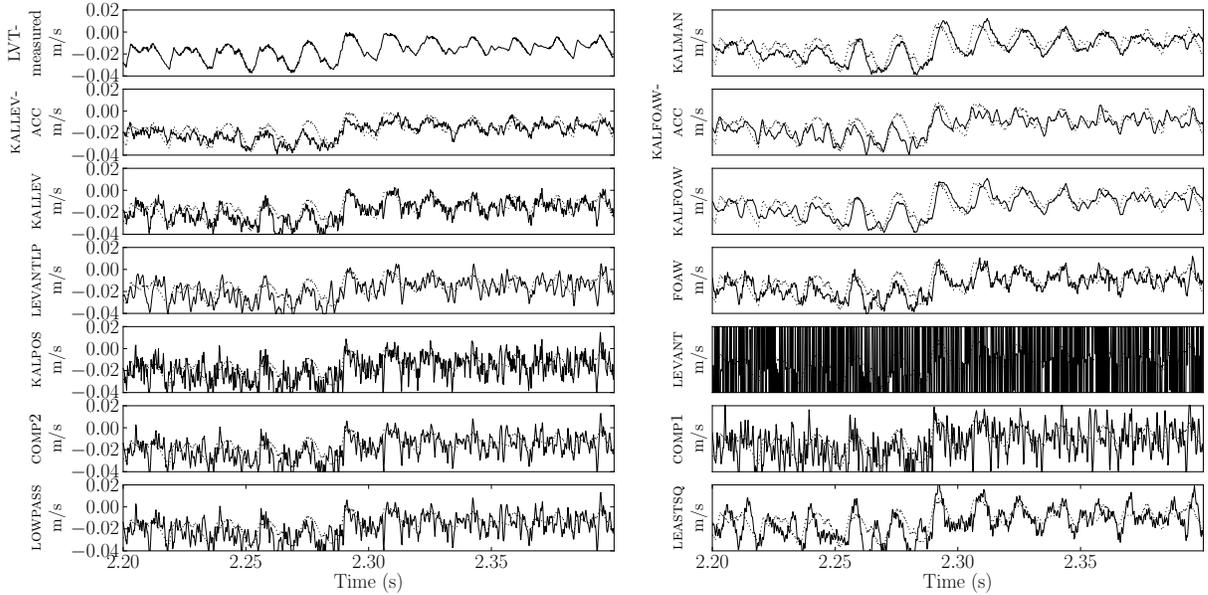


Fig. 6.13 All estimators applied to a section of a bow stroke in recording $F_{\max} = 1.92$ N, with globally optimised parameter sets in use for each estimator, as found in Table 6.2. The LVT-measured signal is overlaid on each plot as a dotted line for comparison.

the FOAW window was maximized for over 98% of time steps, and shortened windows were irregularly spaced in all cases, indicating that indeed the FOAW filter is not reacting to stick-slip transients and thus is acting like a simple sliding window line fit.

Overall, no significant differences could be found for the best 6 estimators on both error and delay criteria simultaneously. Therefore we can conclude that the KALMAN estimator provides the best global results, since it is not significantly different than LEASTSQ in delay, and beats it by a large margin in error performance. No significant improvements could be found by including measured acceleration, nor by means of using the FOAW or Levant methods to improve measured acceleration.

Finally, we note that the least computationally-intensive estimator with good ranking is the post-filtered Levant differentiator, LEVANTLP—it has approximately 90% improvement in error performance on our data set compared to LOWPASS, with similar delay.

An example section of a bow stroke with all estimators applied can be seen in Fig. 6.13.

6.4.6 Gain-specific optimisation

Although a one-size-fits-all solution would be ideal, we wanted to validate against the possibility that certain estimators may be more appropriate for particular gain conditions. We therefore considered how the optimisation results may differ when considering only the low- or high-gain recordings.

The recordings were divided into low- and high-gain groups, where the low-gain group consisted of 8 recordings from 0 to 1.9 N, and the high-gain group consisted of 7 recordings from 2.2 to 12.1 N. In this case, no recordings were excluded from the analysis.

Individual optimisation runs were performed on these two data sets. These results are shown in Figs. 6.14 and 6.15 for low- and high-gain conditions respectively.

In the low-gain results, we see an effective equivalence between most estimators, with the exception of poor delay performance from FOAW and poor error performance from LEVANT. Again we do not see good usage of the accelerometer data, with both complementary filters behaving nearly identically to LOWPASS. Overall, an improvement of approximately 30% on error and 50% on delay versus LOWPASS can be seen in 3rd-order Kalman-based estimators and LEASTSQ.

The high-gain results are perhaps more interesting, in that they show significant benefits of using the hybrid Kalman methods. In particular, the KALLEV and KALLEVACC estimators are the only estimators to beat LOWPASS on both error and delay criteria. Additionally, for KALFOAW, there is a significant improvement from incorporating accelerometer data in KALFOAWACC, though it does not improve on KALLEV performance.

This is demonstrated in Fig. 6.16, where improvements to the Kalman estimation are shown from KALFOAW to KALFOAWACC from simple addition of accelerometer information without modification of the covariance parameters.

6.5 Discussion

The optimisation results show that 3rd-order Kalman-based estimators perform overall better than other approaches on our data set when simultaneously considering error and delay criteria.

Improvements attributed to the use of Levant's differentiator or FOAW to inform the Kalman process are not as clear: in low-gain conditions, there seems to be little benefit to

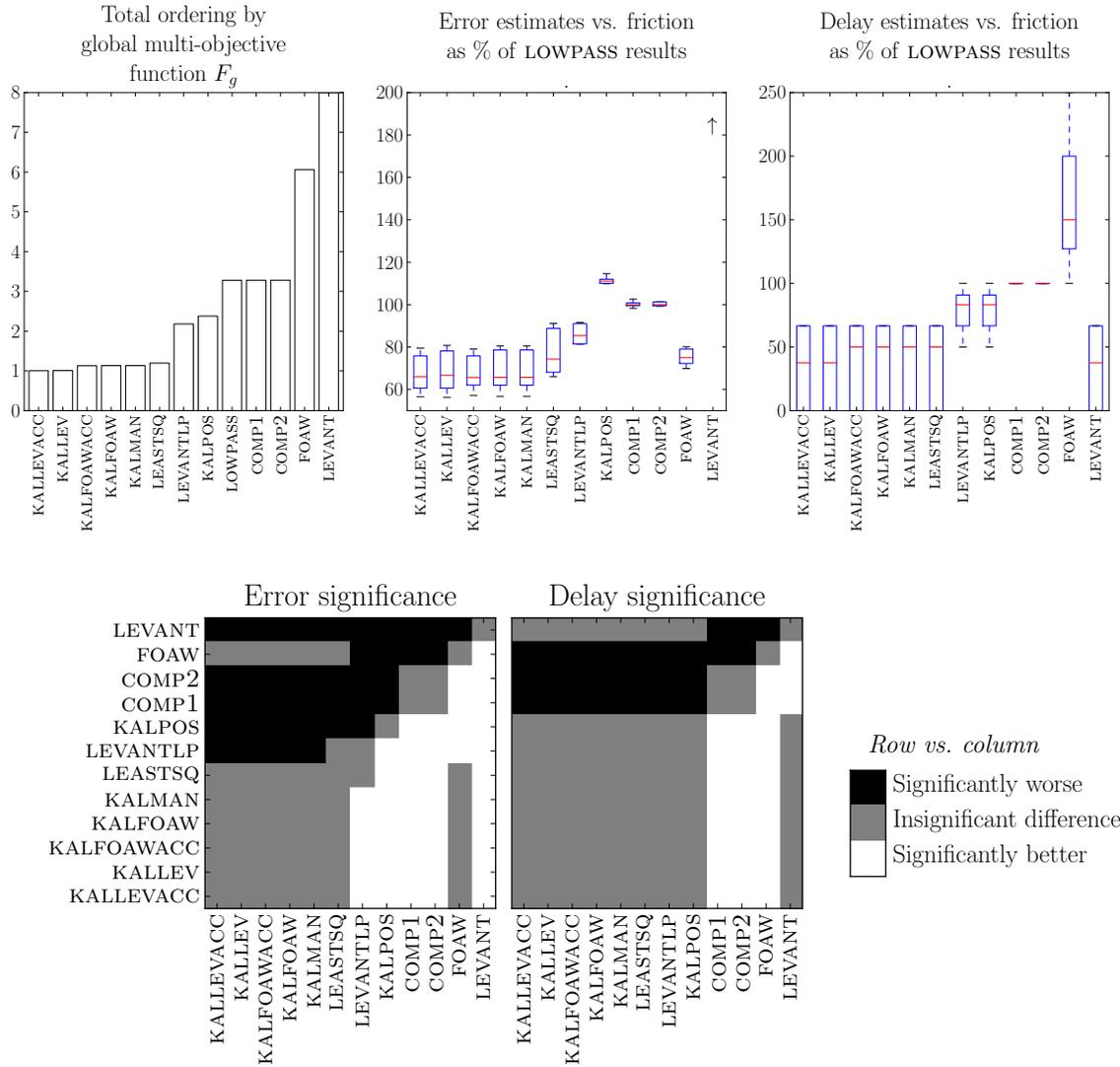


Fig. 6.14 Error and delay evaluation for low-gain recordings, with significance of differences according to Wilcoxon rank-sum test. Here, the significance between estimators is greatly reduced. All Kalman-related solutions as well as the least squares fit beat the low pass filter by approximately 70% on error criterion and 50% on delay criterion.

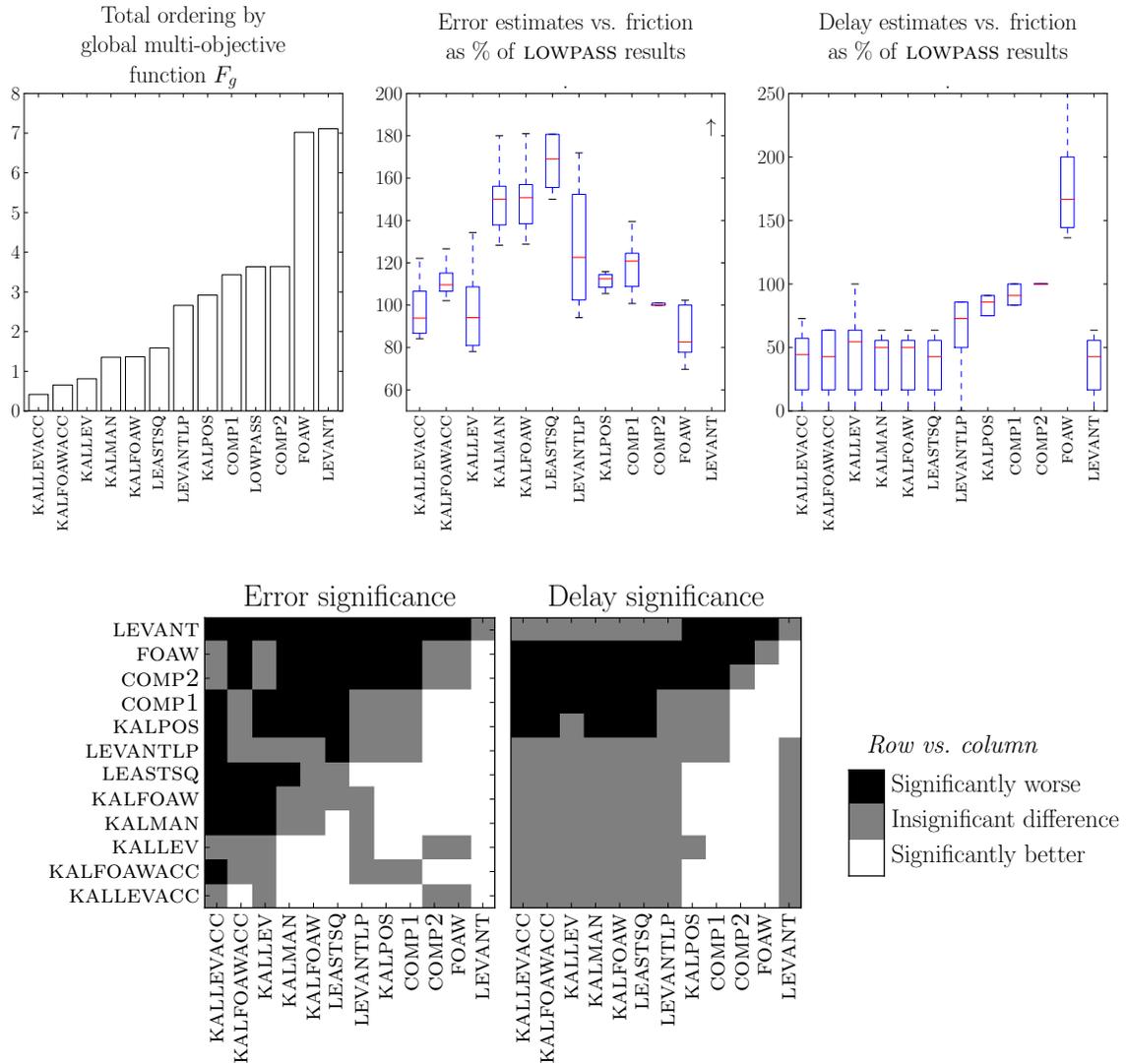


Fig. 6.15

Error and delay evaluation for high-gain recordings, with significance of differences according to Wilcoxon rank-sum test. This time there is a significant improvement in error performance when Levant’s differentiator is used to inform the Kalman filter. Additionally we see some improvement between KALFOAWACC and KALFOAW estimators, indicating some usage of the accelerometer data, although complementary filters are still no better than LOWPASS. We note that the LOWPASS estimator also does surprisingly well in this scenario in terms of the error criterion—only KALLEVACC, KALLEV and FOAW did better. Of course, as usual its trade-off is poor performance in delay, giving it a low global ranking. While LEASTSQ was the among the best for error in the low-gain condition, it performs worst here.

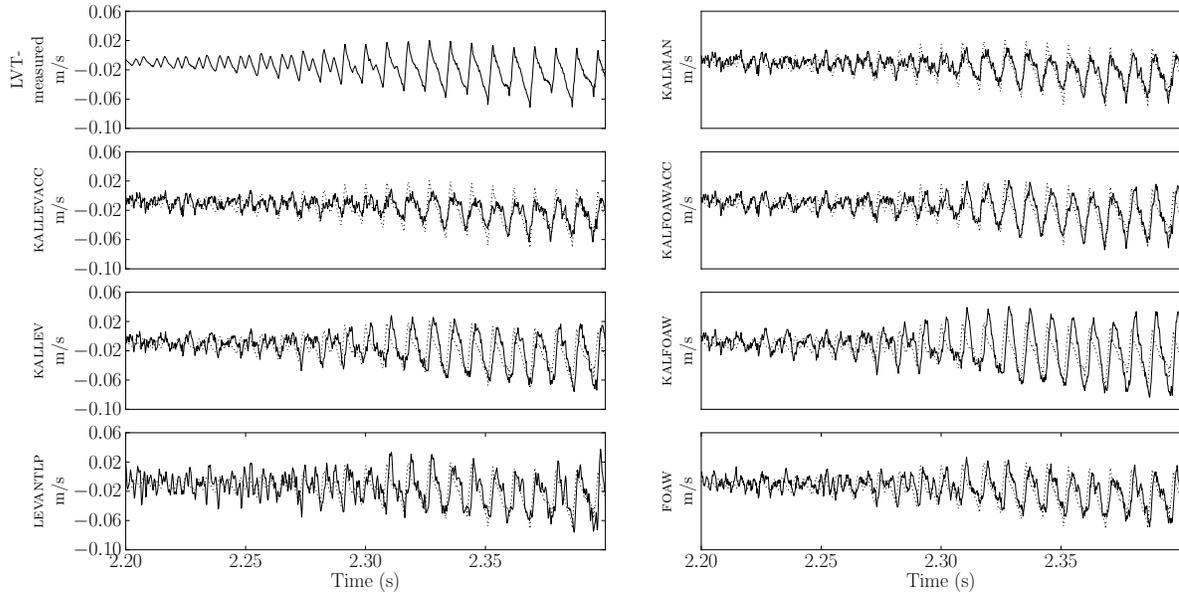


Fig. 6.16 3rd-order Kalman filter estimators applied to the attack of a bow stroke in recording $F_{\max} = 7.67$. Here, the optimised parameter sets are *not* used; rather, we show the results of the same covariance parameters applied in all cases, specifically $Q = 3.4 \times 10^4$, $R_p = 2 \times 10^{-10}$, $R_a = 4.5 \times 10^2$, arbitrarily taken from the optimised parameter set for KALLEVACC. LEVANTLP settings are $C = 100$, LPF = 300 Hz, and FOAW uses a 12-point window with noise margin 0.0001 m. In each plot, the LVT-measured signal is overlaid with a dotted line for comparison. The influence of acceleration information from double-differentiation and from accelerometer measurements is thus demonstrated by comparing the respective results for Levant- and FOAW-based hybrid Kalman methods. In particular we note that adding accelerometer data seems to improve time response at the the transient peaks, in addition to improving overall accuracy, although for KALLEVACC it seems detrimental at the peaks. This example well-reflects the result in Fig. 6.15 that KALFOAWACC improves over KALFOAW, while KALLEVACC does not improve over KALLEV. Note also the relatively poor performance of all algorithms for low-velocity estimates at the beginning of the attack.

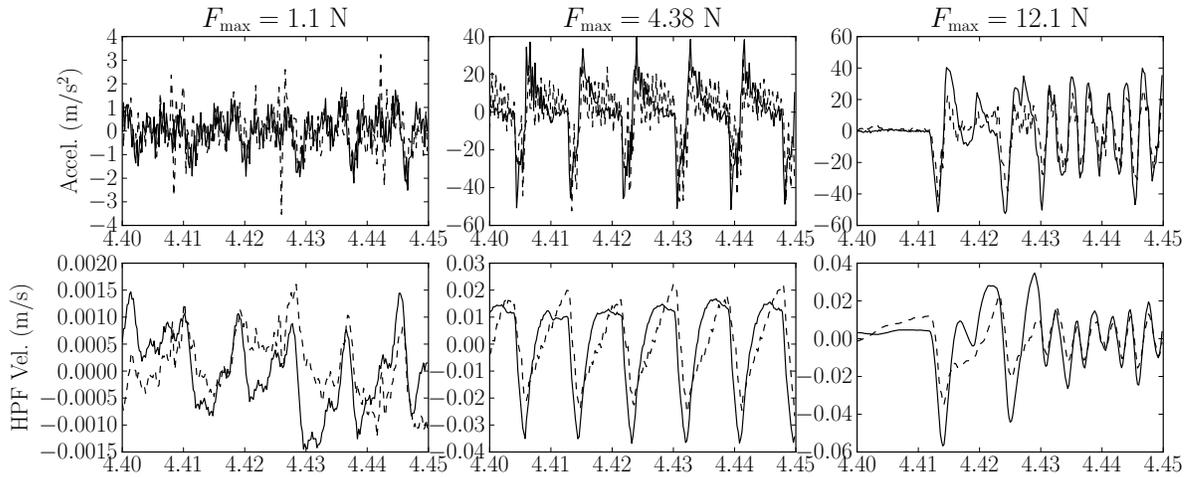


Fig. 6.17 Example comparisons of velocity and accelerometer data. In three recordings, velocity and accelerometer data are shown, with derived signals overlaid. In all graphs, the solid line is the accelerometer measurement, and the dotted line is the tachometer measurement. Therefore, in the acceleration curves, the tachometer measurement has been differentiated, and conversely in the velocity curves, the accelerometer measurement has been integrated. In all graphs, the data have been high-pass filtered at 20 Hz to remove bias for comparison. Although there is clearly more error in the sense of signal-to-noise ratio in the low-gain case, we can see that there is in general good agreement between the sensors. Therefore the rejection of accelerometer measurement during optimisation was unexpected.

these hybrid methods, however in high-gain conditions there does seem to be a significant improvement in error by the use of Levant's differentiator, without incurring a delay penalty.

As for the use of acceleration measurement, it seems to be the case that both complementary and Kalman filters were unable to see benefits from our accelerometer recordings. We do not find it likely that this is an intrinsic fault of these methods, therefore we must assume that this is attributed to an ill-configured electronic or mechanical configuration, or possibly a bad calibration.

This is surprising since we did clearly see the resemblance of the velocity signal when examining integrated accelerometer data, and during calibration the amplitude was matched well to the LVT velocity recording. See Fig. 6.17 for some example comparisons of accelerometer and velocity measurements.

However, one possibility is that although accelerometer data was available, the position

signal measured by means of a good LVDT displacement sensor simply provides higher-quality, bias-free velocity information when properly treated, and therefore the filters preferred to select this source of information. Improvements to our acceleration measurement apparatus must therefore be considered more carefully in future work.

That said, the KALFOAWACC estimator did show an improvement in error performance versus KALFOAW in the high-gain condition, suggesting that accelerometer data was useful in this case. It did not, however, beat KALLEV with significance, which does not depend on accelerometer information. This suggests to us that it is possible that improvements to the accelerometer measurement, if they were indeed faulty, should result in better velocity estimates on both criteria.

6.6 Conclusion

In this chapter we looked at the use of multi-objective optimisation for the purpose of tuning the parameters of several velocity estimators. The resulting parameter sets were discussed, and finally the results were compared for their performance on error and delay criteria.

We found that the best results could be had by a 3rd-order Kalman filter approach, with the 2nd-order least squares fit providing the next-best result for non-Kalman approaches.

We discussed some reasons why FOAW did not perform as well as expected on the delay criterion, namely that the nature of the bowed-string velocity signal is not conducive to the noise-margin test it uses for adaptive shortening of the window length, causing it to mostly act as a sliding-window 1st-order least squares fit.

We did not find significant improvements for the use of hybrid Kalman methods on our whole data set, however such improvements were found when considering only the high-gain conditions. Additionally, we did not find improvements by means of integrating accelerometer measurements.

In the next chapter, we show that improvements to the velocity signal provided an improvement for the impedance bandwidth of friction display, as well as the maximum friction impedance that could be rendered, while maintaining a low noise profile.

Chapter 7

Online Evaluation

In the previous chapter, we showed evidence that several estimation and measurement methods may be able to reduce signal noise while introducing less delay than a simple low-pass filter. As demonstrated in Chapter 3, according to teleoperator theory this should lead to an improvement in the impedance range of the display, while to the benefit of audio-haptic interaction, it should minimize the amplitude of perceived noise in the velocity signal that stimulates the acoustic model.

For bowed string simulation, this means that stronger bow-string coupling can be simulated while keeping the quality of audio and force output within a desired range. To validate this idea according to our results from Chapter 6, it is necessary to perform human testing of these algorithms in an on-line scenario to ensure that the improved velocity signal does in fact lead to improved quality of our interactive simulations.

The approach we have taken for this evaluation is two-fold: firstly, to have the relative noise performance be rated by human operators; and secondly, to have operators determine the friction impedance range while under control of each estimation method. These results were then compared with measurements of the resulting noise amplitude.

The perceptual rating task was necessary because we found that measurement of the noise level alone did not reflect well the relative performance of the estimators in terms of how each type of noise was perceived. We thus additionally compared subjective ratings with the spectral profile of a recording of each estimator noise in order to gain some insight into this relationship.

This chapter describes the evaluation methods and results, followed by a discussion on

the scope of this study for quality evaluation and what more might be done to evaluate interaction with acoustic models.

7.1 Experiment #2

We evaluated the performance of each algorithm under the control of the bowed string model described in 5.1.¹

We used the method of adjustment for this study. This decision was made for multiple reasons: firstly, as described in Chapter 3, we learned during the pilot for Experiment #1 that the number of trials needed for the staircase method is prohibitive when subjects are asked to perform an active, muscle-intensive task such as moving against friction. Since, in the current study, we wished to evaluate many different conditions, the number of trials would be multiplied.

We also learned that the task of identifying distortion is somewhat difficult for subjects when presented with a statically-parameterized condition. It seemed to be much easier for subjects to understand what they are intended to identify when given direct control over a continuous parameter. Rather than spending time considering the potentially subtle differences between two conditions, when subjects can actively explore the parameter space they tend to internalize the effects of the adjustment with a short amount of practice, and are soon able to almost immediately find the point of interest within the range; this makes each trial take much less time than for a forced-choice approach, and, we found, seems to lower the variability in the results.

Furthermore, the parameter space itself can be problematic for static examination. It was noticed that an apparently stable system above the limit can be destabilized, and subsequently if the gain is reduced it may continue in an unstable regime unless the gain is reduced further and then returned to the former location. In other words there is some form of hysteresis present in the effects of marginal stability, since an oscillating state encourages duty cycles to continue rather than settle down automatically. This behaviour seems much easier for a human to understand through active exploration, than by being presented with choices of stable and unstable systems.

¹This study was performed with the approval of McGill's Research Ethics Board, REB #105-0908.

7.1.1 Stimulus

For the rating component of the task, subjects interacted with the bowed string model running at 5 kHz on the TGR hardware (see Appendix A) by moving the handle back and forth along its single free axis, and listened to the sound of the string as well as the background noise produced by different methods of position differentiation. Friction forces were exerted with a constant $F_{\max} = 1$ N. Subjects could control which velocity estimator was in use, actively switching between them to compare quality in order to assigning a normalized rating to each.

As stated, for the second part of the experiment, the objective of the study was to determine the upper range of friction impedance display for each method. We characterise the impedance range for the estimator as the maximum displayable friction gain, F_{\max} such that the display remains stable.

As in previous chapters, we do not speak of friction coefficient, since we used only one degree of freedom, and the orthogonal axis of force was not considered, therefore we discuss the friction model only in terms of the maximum force amplitude allowed to impede motion in the free axis. This corresponds to the height of the corner of the friction curve forming the rosin model, described in Chapter 5, Fig. 5.2. The other parameters of the force-velocity function, the width of the viscous portion, and the curvature of the sides, were held constant, and therefore by adjusting F_{\max} , subjects controlled the slope of the viscous relation $F_b = -\mu F(v_\Delta) \cdot v_\Delta$, such that $\max F_b = F_{\max}$, where v_Δ is the velocity difference between bow and string and F_b is the friction force on the end effector.

Subjects were told to find the margin of stability, which was obvious due to device oscillation. Typically this was easily identifiable not only by vibration of the device, but by a strong distortion in the sound of the model, however, some conditions such as the unfiltered Levant differentiator could experience significant vibration while still capable of producing the string sound.²

Therefore subjects were encouraged to use the breakdown of the string behaviour as a cue to help tell the difference between stable and unstable conditions. Specifically, they were told to “find the point immediately before the model is no longer controllable as a bowed string, and no bowed string sound can be heard or felt.” This coincided with

²Note that although it was clear a priori that this model was too noisy to be of use, we were interested in how its exceptional delay performance would translate into impedance improvements.

the F_{\max} where significant low-frequency vibrations could be produced due to marginal stability. A few training trials were used to demonstrate the phenomenon.

Results therefore indicate the inner boundary of stability, where the string sound could still be produced, but a small increase in F_{\max} would lead to instability and a breakdown of the model’s natural acoustics.

7.1.2 Conditions

The estimators used in this experiment were the subset from Chapter 6 that depend only on the position signal: these were LOWPASS, LEASTSQ, FOAW, LEVANT, LEVANTLP, KALPOS, KALLEV, KALFOAW, as well as LVT, which was a direct usage of the tachometer signal.³

Estimators were executed concurrently on the Toro hardware, in order to remove any initial convergence time that some algorithms require, so as to allow quick switching from one condition to another. Some algorithms were too intensive to run concurrently, and had to be switched. These included the different 3rd-order Kalman filters, which were selected by changing the covariance parameters of a single function call, rather than by running dedicated filters concurrently, and the LEASTSQ and FOAW estimators, which were disabled when not in use. Initialization effects after switching were not noticeable.

Under these conditions, we were able to run the system at a maximum of 5 kHz, with the exception of KALFOAW, which was only able to run at 4 kHz, and therefore had to be executed separately.⁴ For this reason, KALFOAW was left out of the rating task, since subjects needed to be able to switch quickly between conditions in order to compare them—time required to reload the firmware and reinitialize internal states made it prohibitive to change the sampling rate during the experiment. For the impedance task, rather than reject the estimator outright, KALFOAW was evaluated alone after all trials of the other conditions. Therefore it was not included among the randomized presentation order. However, since subjects were evaluating overt device behaviour, as opposed to testing subtle perceptual

³We unfortunately were not able to include accelerometer-based methods here due to equipment availability, however this was already a large number of conditions for subjects to rate simultaneously.

⁴An option would have been to run the whole experiment at 4 kHz, however this would have required re-recording gesture data and re-optimising all algorithms for this condition, since recordings had taken place at 5 kHz; time constraints did not allow for this. Additionally, since 5 kHz was already a compromise in terms of audio rendering quality, we did not wish to further reduce the sample rate for all conditions. Prior to realizing the computational limits, we had intended to perform this study at least at a rate of 20 kHz, to allow a more natural-sounding bowed string; at 5 kHz, the sound is clearly degraded, but recognizable.

properties, we do not find it likely that this discrepancy meaningfully affected the results.

7.1.3 Apparatus and Methodology

Subjects were seated in front of the haptic device and told to grasp the handle with their right hand, while their left hand was used to adjust the MIDI knobs.⁵

In the first part of the experiment, subjects were presented with eight on-screen sliders, and these were associated with eight MIDI knobs which were labeled in the same order. These knobs did not modify any model parameters—they were used merely for rating the conditions. However, moving a knob resulted in the associated condition being selected.

Subjects were asked to first listen and manipulate each condition, paying attention to the noise qualities, and then to set the noisiest condition to the maximum value, and the least noisy condition to the minimum value. Subsequently, they were asked to rate the noisiness of the other conditions according to their subjective position on this relative scale. They were given as much time as they wished to listen, compare and rate without interruption. In general this lasted under 10 minutes, at which point the subject would indicate to the experimenter that they were finished.

In the second part of the experiment, subjects were given control over a single MIDI knob controlling F_{\max} . Each trial started with a friction level of zero, and subjects were asked to increase the logarithmic gain by turning the knob until discovering the point at which the model began to self-oscillate, fine-tuning it by compensating hysteresis if necessary. Throughout, they listened to the sound of the bowed string model in a pair of headphones.⁶

7.2 Results

The results of the friction gain adjustment portion of the experiment are given in Fig. 7.1, and the rating task is analysed in Fig. 7.2.

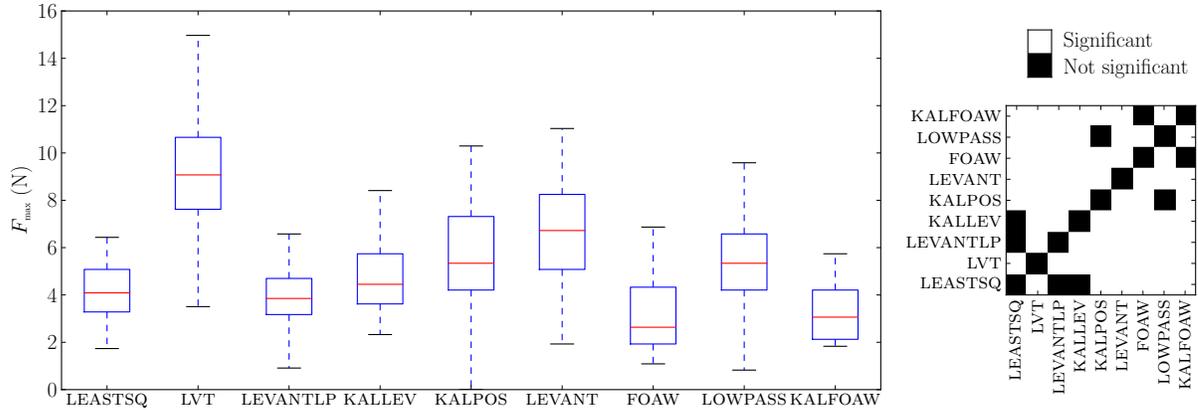


Fig. 7.1 Maximum friction force F_{\max} for bowed string interaction. Significance is determined by Wilcoxon’s rank-sums test, with $p < 0.05$ indicating confidence that distributions are not the same. We see that LEVANT performs best after the tachometer, agreeing with the hypothesis that delay is a significant factor for the impedance range, however the next best estimators are KALPOS and LOWPASS, which did not perform exceptionally well for delay. Therefore we must assume there are additional factors influencing performance at high gain. Note that due to technical limitations, KALFOAW was executed at 4 kHz and tested independently, while all other estimators were executed at 5 kHz and their order was randomized during testing.

7.2.1 Maximum impedance judgement

We notice firstly that the tachometer, condition LVT, features nearly twice the impedance range as any estimator. Secondly, although it was very noisy, LEVANT, the non-filtered Levant differentiator, performed second best, agreeing with the hypothesis that delay, independent of noise, is an important factor for the range of stability. Confirming this, the FOAW estimator performed worst, and was also a weak performer on the delay criterion on Chapter 6.

However, the next-best impedance ranges after LEVANT belong to KALPOS and LOWPASS, which did not perform exceptionally well on the delay criterion in Chapter 6. This is rather confounding, as it suggests that there are other factors affecting behaviour at high-gain.

⁵Akai LPD-8, providing a 7-bit potentiometer.

⁶Bose QuietComfort 15 noise-cancelling headphones.

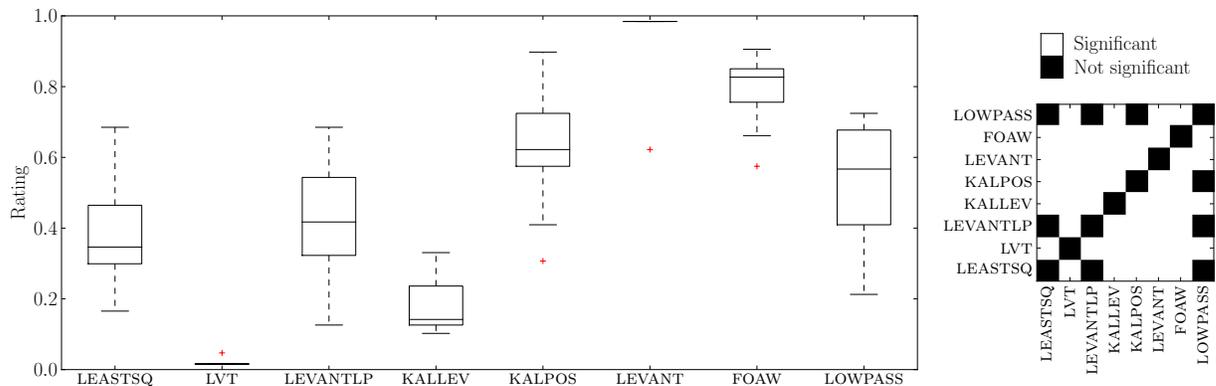


Fig. 7.2 Subjective ratings of noisiness for several estimators. Significance is determined by Wilcoxon’s rank-sums test, with $p < 0.05$ indicating confidence that distributions are not the same. The small deviations of LVT and LEVANT are due to instructions that the noisiest and least noisy condition be maximized and minimized within the range—the choices of best and worst were unanimous. Notably, the KALLEV estimator has the best performance after the tachometer in terms of perceptual noisiness. The KALFOAW estimator could not be included in these ratings due to technical limitations.

7.2.2 Subjective noisiness ratings

The subjective ratings, Fig. 7.2, are more easily interpreted. The best and worst performers for subjective noisiness, expectedly corresponding to LVT and LEVANT, were selected unanimously.

The preference for KALLEV is indicated clearly, which confirms its performance on the noise criterion as indicated by numerical evaluation in Chapter 6.

On the other hand the weak subjective performance of FOAW is inconsistent with those results, which showed that it is competitive with the 3rd-order Kalman filters on the global noise criterion. However, despite beating LOWPASS, it was nonetheless ranked with KALPOS and LEVANTLP as a weak performer for the low-gain conditions discussed in 6.4.6, which included the gain we used for subjective testing, $F_{\max} = 1$ N. In that light, the poor performance of LOWPASS here is perhaps more confounding than that of FOAW; however, its rating variance was quite high compared to the other estimators.

7.3 Further investigation of perceived noise

To help understand the perceptual aspects of the noise performance of each estimator, we recorded a stationary signal under each condition. A spectral analysis can be seen in Fig. 7.3, and a distribution analysis can be seen in Fig. 7.4.

In the former, we see that the rankings of estimators are somewhat different in the low- and high-frequency ranges of the spectra. Although not a perfect match, it can be noticed that the high-frequency ranking corresponds better to subjective noisiness ratings, whereas ranking at DC corresponds somewhat better to maximum impedance estimations.

The distributions given in Fig. 7.4 do not seem to provide much information except to show a surprisingly good error distribution for KALFOAW. The improvement in accuracy for this method over FOAW indicates that such a hybrid method may have significant advantages over application of a basic estimator alone.

We do not claim that these observations are important for interaction, since they likely do not reflect distributions during a stroke. However, the correspondance with our impedance test does seem to indicate a tentative explanation of the mismatch between impedance range and delay performance—since our numerical optimisation procedure reduced delay and error analysis to a scalar value, distributions and frequency-dependent characteristics were ignored.

We note that stationary noise was very likely a strong component in subjects' ratings of estimator noisiness, since they were not forced to keep the end effector moving, but rather were encouraged take a somewhat holistic approach, critically listening to both stationary “background noise” as well as noise during interaction.

7.4 Discussion

Although our analysis of stationary noise seemed to help explain some of our observed inconsistencies, it would be more ecological to include an analysis of noise during bow strokes. This is particularly relevant in light of “noise-gating” methods such as the TWO-POINT model described in 4, which allows to remove stationary noise, so that only noise during interaction is still an issue.

However, a decomposition of noise characteristics during bowing is more difficult, since not only must the noise be separated from signal, not easy in the presence of a high-

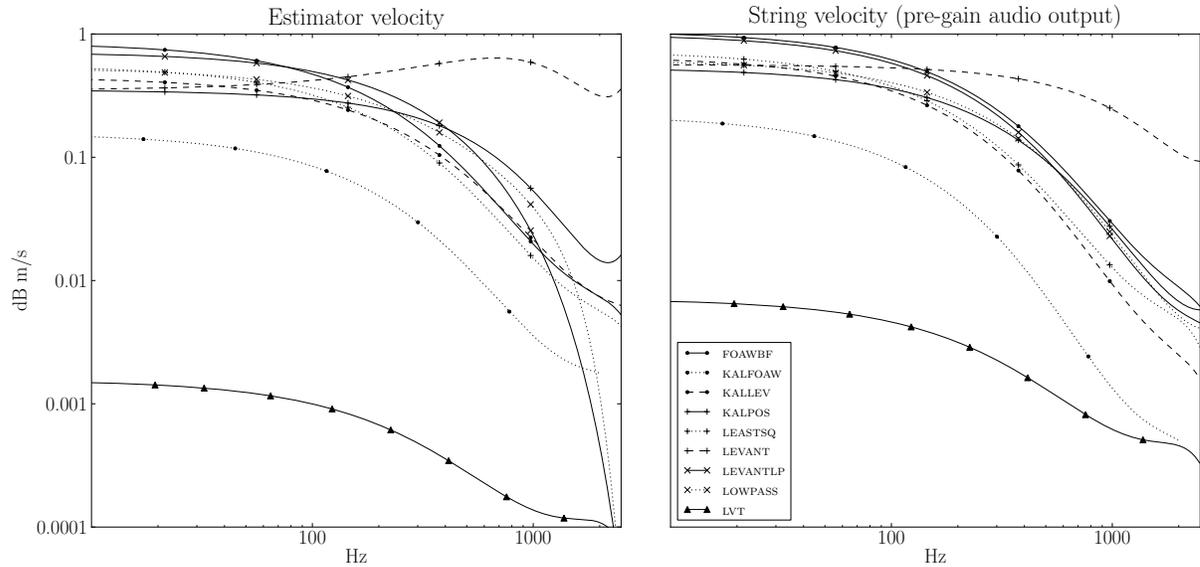


Fig. 7.3 Spectral amplitude of stationary noise for several estimators. The handle of the device was held unmoving at a friction level of $F_{\max} = 1\text{ N}$, as the estimated velocity as well as resulting string velocity from the bowed string model was calculated online. The spectra, determined by FFT after application of the Hann window, have been denoised by spline interpolation for clarity. The slight up-turn of KALPOS and LEVANT are artifacts of smoothing. We see that after the tachometer, which has exceptional noise qualities, the next best estimator is KALFOAW and then, in the high-frequency range, LEASTSQ, followed by an effective tie between LEASTSQ and KALLEV. At DC, however, KALFOAW is followed by KALPOS and then LEVANT and KALLEV, which would seem to correspond somewhat, though not exactly, with impedance performance in Fig. 7.1. Note that KALFOAW was recorded at 4 kHz, while all other signals were recorded at 5 kHz.

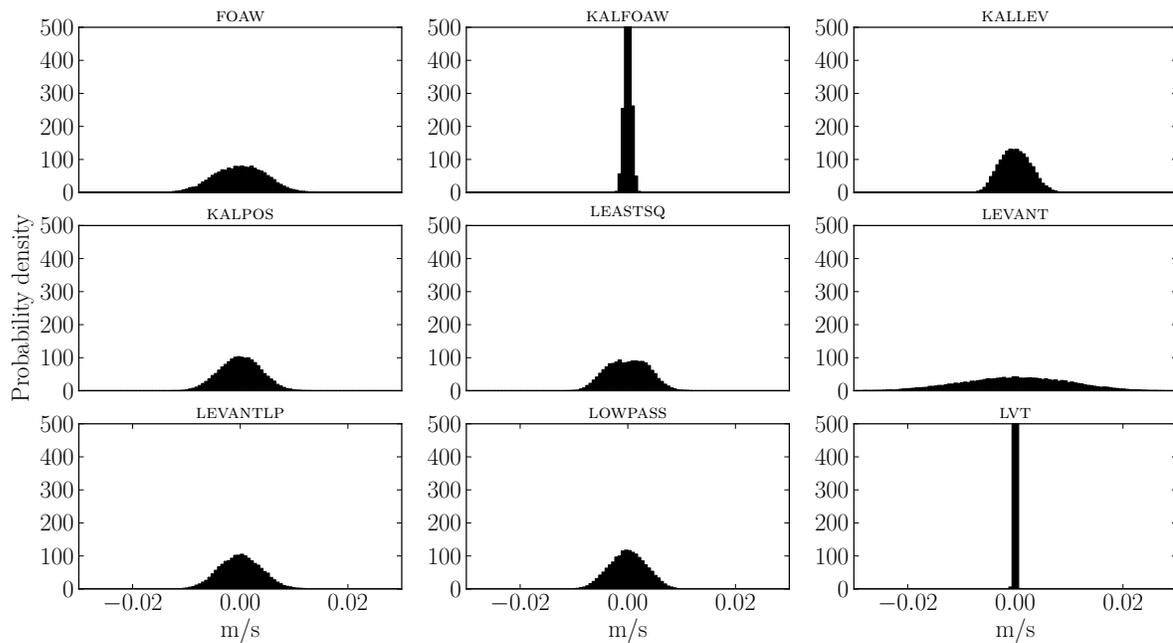


Fig. 7.4 Probability distributions of stationary noise for several estimators. Histograms normalized such that the area is equal to 1 m/s gives the probability density function (PDF) of each noise signal. All estimators feature a normal distribution for stationary signals, however the standard deviation is clearly significantly smaller for KALFOAW and the tachometer (LVT), whereas LEVANT without low-pass filtering features a very wide deviation compared to other estimators. KALFOAW was recorded at 4 kHz, while all other signals were recorded at 5 kHz.

bandwidth response like the bowed string, but we informally observed, and several subjects also indicated to us, that the presence of the string sound seems to partially mask the noise in practice. Finally, real bowing in fact features some amount of noise due to a natural stochastic distribution of imperfections in bow hair [89], making the acceptable qualities of noise during bowed string interaction uncertain without further research. Therefore, such a study must be left for future work.⁷

Nonetheless, these results show exceptionally good performance for direct measurement using the tachometer, and also show that some estimators are indeed preferable. In particular, the combination of the Kalman filter and the Levant differentiator seems to allow good noise rejection while also being competitive in impedance range.

Finally, it is clear that a 200 to 300 Hz low-pass filter actually ranks fairly well in practice, though it is not the best method.

7.5 Conclusion

This study examined the upper limit of friction force, in order to evaluate an increase in impedance range, and also judged the perception of noise at the same parameters, in order to judge whether noise-delay trade-offs determined by the optimisations of Chapter 6 could pay off in terms of overall performance.

We did not directly examine the *lower* boundary of perceived noise, as we did in Chapter 3. One reason is that stability is a very obvious phenomenon that can be easily determined by a subject. Conversely, it may be overly nuanced to ask subjects to distinguish between distortion due to device control vs. inherent characteristics of a sounding model that they are not familiar with, particularly because we know a priori that the model in use already contains several approximations, and cannot be assumed to accurately simulate real bowing.

This is also the reason we did not consider it necessary to strictly use musicians as sub-

⁷In this work we were not interested in psychometric comparisons between our model and a real instrument, but rather we used the bowed string interaction as a scenario for evaluating the performance of noise rejection methods, thus noise rejection as its own goal was considered adequate. Secondly, if an instrument featuring natural noise is to be modeled, it can be argued that it is better to initially remove any unwanted system noise before adding modeled imperfections. On the other hand, it could be possible with some effort to include a *noise shaping* component in the implementation, but this would likely be a strictly more difficult approach and very dependent on specific noise characteristics of the hardware in question, a situation best avoided.

jects: those very familiar with bowing may have expectations that could lead to confusion about quality because of imperfections in the digital model. Although the impressions of musicians would be useful for evaluating the model, this was not the goal of our study.

Furthermore, such absolute perception thresholds would of course depend on characteristics of the transducers: the device, the loudspeakers, and the corresponding gain settings. Therefore the exact thresholds may hold little value since they would be difficult to generalise, but a ranking of each type of noise in terms of perception for both passive and active cases can be a useful tool when selecting an estimation method.

An alternative method for parameter optimisation could have been to establish requirements in the form of an inequality with regards to delay or noise behaviour, and to optimise based on the opposite criterion. Similar to the method of adjustment, it would be tempting to give subjects direct control over the parameters of each estimator and allow them to discover the preferred parameter set. However, adjusting multiple controls while simultaneously judging multiple criteria at opposite ends of the perceptual spectrum (very small distortions and very large instabilities), while requiring active manipulation of the device would seem to be prohibitively complicated for subject-based methodology.

In addition, some estimators may pass through unstable regions of the parameter space, therefore such direct manipulations could become dangerous or, more likely, lead to numerical errors that require system reset. Such possibilities would need to be understood and protected against.

Such a process could be made more efficient if methods can be developed to automatically perform perceptually-informed qualitative judgements of stability and noisiness. In virtual wall stability, automatic testing and simulation are sometimes used, e.g. by loading the end effector with an elastic band and counting the number of bounces [50]. Such methods allow for automatic testing of a large parameter space without the overhead of involving human decision-making in the loop.

However, acoustic behaviour mixed with high- and low-frequency noise is more difficult to judge automatically with similarly simple heuristics. Therefore, signal processing techniques such as distribution and frequency analysis as we have presented, perhaps combined with machine learning methods, should be designed in future research to classify noisy acoustic signals according to human judgement.

Chapter 8

Discussion

In this chapter we summarize thoughts and observations regarding the work presented in this thesis, and present some general discussion on the topic of audio-haptic simulation.

8.1 Generalizability of results

As mentioned, we chose to pursue evaluation and improvement of velocity estimation methods because we felt that the position-dependent friction approach applied to the digital waveguide was not easily generalized. Strictly speaking it may be possible: effectively the approach consists of an inversion of the typical design of an impedance haptic loop, which we describe in the following subsection.

Conversely, we have assumed that improving the velocity signal is the most general approach, however there are some caveats for the applicability to common musical interaction, which will be discussed below.

8.1.1 Generalizability of DISTPLUCK

An impedance device takes position and its derivatives as input, and the usual approach is to develop a penalty force in response to constraints on position and velocity which synthesize virtual effects such as walls, textures, viscosity, etc. Often this is mediated by a virtual coupling which maintains passivity by placing limits on maximum demanded impedance [31]. In the DISTPLUCK model (chap. 4), rather than driving the excitation using velocity, we instead utilized a friction model to stimulate forces on the instrument.

The resonator motion is conversely used to modulate the friction appropriately.

Therefore, rather than device forces being commanded by the model reaction force, potentially mediated by a virtual coupling, instead the impedance device executes a dissipative interaction that is modulated by the model. This dissipation itself is not unique and can be implemented via velocity-dependent relationships, but the expression of the friction model in terms of position differences makes it robust to noise. From the model point of view, then, forces are not applied to the end effector, but rather *friction* is commanded by control of the anchor point.

Thus the impedance-device-plus-anchor can in a sense be thought of as an admittance—force is “sensed” through the spring, which either admits motion of the point or not, based on constraints which implement a friction behaviour. This is fundamentally different from a virtual coupling which accepts a force signal as input and transforms it before passing it on to the device motors. The question remains as to whether such a principle can be used to drive any arbitrary friction interaction.

We believe it should be possible to extend to some class of other models, however the main difficulty is that if we desire anything other than smooth friction, the model must be able to affect the anchor point in order to modulate friction. This was achieved in DISTPLUCK by considering the bow-string tension as related to the device-anchor potential and thus to friction force; this is rather model-specific, and a generalized approach is certainly desirable.

We think the solution is likely to be along the lines of a block that takes friction command as input and outputs information such as force exerted, energy dissipated, etc. If interaction models could be designed around such an interface, this would allow an arbitrary friction model to be “plugged in,” and a choice of position-based friction would be responsible for transforming the commanded friction into a position difference and stiffness. Other choices could implement friction via a hardware damper circuit or actuator [51], but critically these choices must allow for detection of exerted friction force.¹ However, for the moment we have left further development of this theory for future research.

¹Note that this could be implemented via velocity tracking! However this only brings back the original problem.

8.1.2 Applicability of velocity estimation improvements

Although the utility of improving the quality of any measured signal is clear, we note that in considering the computational costs of advanced estimation techniques it should be noted that these improvements do not globally apply to any interaction. Rather, we have established that their main effect is to extend the range of stable friction impedance. As mentioned in Chapter 7, it does not necessarily follow that these improvements have a perceptible effect in the lower parts of the impedance range. For example, we have informally tried to blindly tell the difference between several estimators configured for low-noise, the tachometer, and an 80 Hz low-pass filter at low friction, and it seems almost impossible to do so as long as noise is not apparent and there are no limit cycles.

If such impedances are not required, it may not be necessary to employ costly estimation methods. We mention this because we noted during this work that in fact the upper impedance range does not actually apply to many typical bowed string scenarios.

Although we have not measured friction in real bowing for comparison purposes, the literature suggests that low-density, taught strings such as on a violin do not exert more than approximately 1 to 2 N of friction force for the normal, playable range of bow pressure and contact area, see 3.4.1. In this case it may be perfectly adequate to use an aggressive low-pass filter to eliminate perceptible high-frequency noise. Moderately higher forces however are experienced on the cello and double bass. We can also imagine other musical interactions, e.g. rubbing a drum membrane with a rubber mallet, which likely involve stronger viscous or frictive coupling. It would be a useful exercise to characterise friction levels found in a variety of musical interactions.

Since a main point of simulating this interaction would seem to be the ability to switch between such scenarios, it seems nonetheless useful to maximize performance in as wide a range of simulations as possible, even if it is not applicable to typical violin bowing. Moreover, if no one-size-fits-all solution can be found, knowing when it may be appropriate to use one estimator over another is useful knowledge.

8.2 Observations

8.2.1 Perception of noise in the presence of an acoustic model

During development of our experimental methodology, we found that close to the perceptual threshold, users may have difficulty telling the difference between technical deficiencies and their expectations of model behaviour. For instance, comparing two similar physical models may be challenging since not only will the time-domain behaviour differ, but additionally the model reaction to signal error may influence perception of the model, such that conclusions may be biased by technical aspects.

Although model comparison was not discussed in this thesis, this was one problem that we encountered in collaboration with Florens [119] while trying to determine how to study the differences between the CORDIS-ANIMA model [43] and the digital waveguide approach; since in the former case the noise is filtered, and in the latter, it is emphasized, it became difficult to draw conclusions about the models themselves when the responses to signal error were so obviously different. Moreover we suspected that the two models might converge if more processing power could be brought to bear on the simulation, in order to simulate higher modes, but increasing the coupling gain and the frequencies of the CORDIS model also increased the apparent noise, resulting in an overall degradation. Thus comparing models against each other or against a gold standard of the real thing is made difficult during interaction due to noise.

One approach to this problem could be to show formally that a model never accentuates noise, but only diminishes it. Then, it follows that it is sufficient to diminish the noise *source* below the threshold of perception, to claim that model behaviour is independent from noise. This would decouple model behaviour from technical problems and would allow more valid comparison of models. Of course such a criteria may simply not be true for many acoustic models that feature non-linearities or high-frequency resonances.

We found in our pilot of Experiment #1, Chapter 3, that users sometimes had difficulty even identifying distortion, since they sometimes assumed it was a property of the simulation rather than an undesired effect due to error. In the case of noise, some users interestingly did not identify it, despite clearly being able to feel it, since they assumed it was something to be ignored. This is an effect similar to how a listener might no longer notice background noise in a recording after a few seconds, but asked to pay attention he might nonetheless be able to correctly tell you whether it is present.

From a certain point of view, this coupling between model behaviour and perception of error could be considered a good thing, since it may help to mask signal-related problems. However it does make objective study more difficult. We suppose that such difficulties may contribute to problems of the “uncanny valley”, as previously mentioned in this context by O’Modhrain [89]—that users may have the impression of something close to reality, but with some remaining distinguishing distortions that cause them to feel like there is something wrong, unable to explain the issue in concrete terms.

8.2.2 On device-in-the-loop acoustic interaction

Although, as haptics researchers, we often consider improving the quality of interaction to be the foremost goal of our work, another point of view is that the role of interaction is to help mediate between a performer’s intentions and the resulting product—the audio, or other media which is to be displayed or recorded. Hypotheses in this case are, firstly, that haptic feedback can help the performer more easily or closely achieve this goal, and secondarily that as haptic feedback is improved to more closely approximate real mechanical interaction, it will help even more.²

If haptic interaction with acoustic objects is considered as a means to an end, i.e. for production of media content, then it is mandatory to consider the effect of the device itself on production. The audio synthesis is performed by a digital mechanical simulation, of which the device is made part. Thus, characteristics of the device, such as inertia, friction, and signal quality, do not only impress upon the quality of the operator’s experience, but actually modify the resulting sound. See Fig. 8.1 for an example of the effect that force feedback can have on the sound of a bowed string model.

This is an important consideration since it means that non-idealities of a particular device, so-called “parasitic forces,” do not merely degrade the operator’s experience, but rather an inability to properly eliminate or compensate them may limit the practical utility of acoustic interaction on the whole. In short, *two devices implementing the same acoustic model at the same sample rate may not produce the same sound*. The notion that the device and electronics must be considered more and more like a high-quality transducer similar to the properties of a loudspeaker in audio [56] is strongly reinforced.

²Note that this is not at all a given. O’Modhrain [89] found that any form of haptic feedback tended to help in a musical task, but one type did not help more than another.

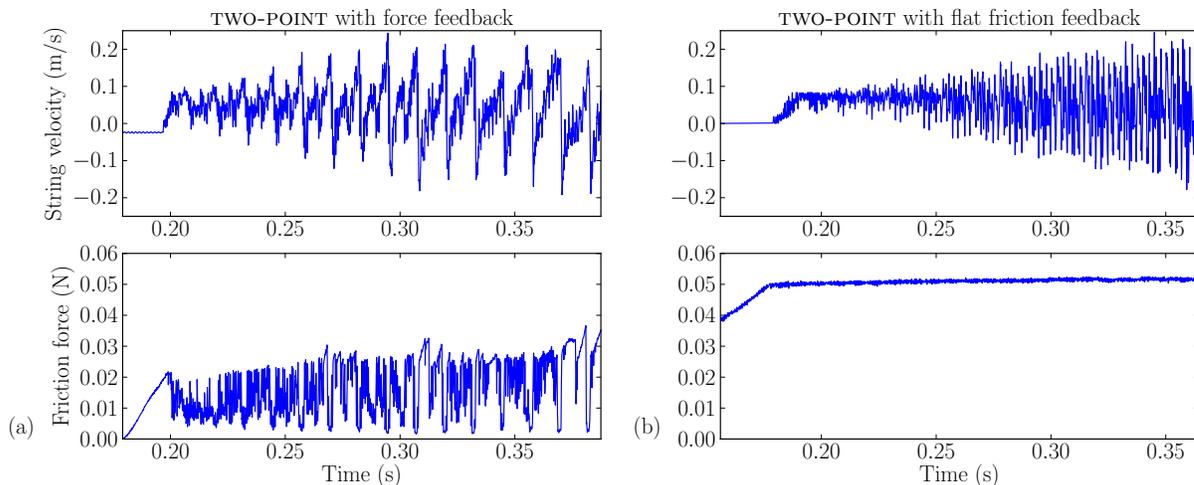


Fig. 8.1 Two close-ups of the attack of the TWO-POINT model of similar gestures. (a) With bow-string friction feedback. (b) With smooth friction force feedback. It can be seen that the presence of model feedback force modifies not only the feeling (lower), but also the sound (upper). Therefore the device’s high-frequency behaviour must be considered as influencing the sound.

Fortunately, it is possible through various means to characterise the limits of a device as to its potential for acoustic interaction. For example, we described methods in Chapter 3 to analyse the performance of inertial simulation by means of virtual coupling. If virtual inertia could be simulated perfectly within the relevant bounds, then we should be able to claim that inertial effects on the sound will be non-existent.

Noteworthy, however, is that the perfect simulation of inertia was *not* possible. Inertia simulation was successful only within a part of the frequency range, which could be expanded by increasing the coupling stiffness, but this was limited by stability concerns and noise amplification. This shows that while inertia simulation might have a convincing effect on the gesture level for the operator, acoustic vibrations that are in the higher frequency range will remain affected by the device’s inertial property. To claim that inertia has been perfectly compensated, it would be necessary to show that the frequency range achieved by the coupling stiffness exceeded the interesting range for audio feedback, a rather more difficult proposal than for haptics alone, but perhaps achievable.

Similar methods may be used for friction compensation and other device characteristics, if it can be proven that they are compensated within the audio range. This might be difficult to parameterize however since non-linearities like friction have characteristics that are not

frequency-dependent. Nonetheless, methods for friction compensation such as acceleration feedback exist [7] and this is an active research area in control.

8.3 Conclusion

While audio-haptic interaction shares many of the issues often encountered in haptics and force feedback research, it also encompasses some unique challenges that arise from a different set of requirements.

Since we are considering a secondary output to a loud-speaker, issues of high-frequency open-loop behaviour become important, such as the perception of broadband noise. Closed-loop issues also arise in the form of friction impedance stability, which for certain interactions may be important, although it is also interesting to consider that there are limits to the impedance needs of many musically-relevant interactions. However, high-frequency behaviour also becomes a closed-loop problem because device characteristics such as high-frequency resonant modes are given the ability to influence the dynamics of the acoustic model. If the goal is to produce ecologically valid interaction and output, this may pose problems.

Use of acceleration feedback and advanced modeling may be able to compensate within a certain range, however the mechanical frequency response of most haptic devices is likely not adequate to compensate far into the acoustic range without distortion [24]. The use of end effector-local tactile actuators may be one possible solution for high-quality presentation of high-band response while utilizing device motors purely for low-frequency forces [49]. On the other hand, a valid alternative approach is to develop interactive synthesis methods such as DISTPLUCK that are mostly insensitive to device-specific behaviour.

Chapter 9

Conclusion

In this dissertation we have looked at some aspects of the application of force feedback simulation to audio-haptic interaction with digital acoustic models. The problem of noise being introduced into the resonator model due to velocity estimation was recognized and investigated. In particular we have concentrated on models of the bowed string, because it is a good example of a constant-contact musical interaction that is already well-researched and has previously been used in this context.

Since the bowed string is a friction-based interaction, it is affected by the quality of the velocity signal, which is typically derived from position measurement, and can therefore in practice be quite noisy. Without special consideration, high-frequency noise can make its way into the acoustic simulation, reducing the overall quality of both sound and force feedback. Although it can be filtered, this reduces the bandwidth of interaction, ultimately reducing the maximum displayable impedance.

We have explored some solutions for reducing the effect of noise specific to the bowed string digital waveguide, which involved re-expressing the device-model junction in terms of a position-based friction approach, avoiding the need to differentiate.

We also discussed more general methods of noise reduction, by means of improving estimation of the position differential as well as incorporating additional measurement apparatus. In this latter case we proposed that perceptual methodology should be used in order to determine when the noise level can be considered below the human detection threshold, however we mentioned difficulties in achieving reliable and generalizable results due to the coupling between transducer apparatus, model behaviour, and subjective preference.

9.1 Review

In this work we have given attention to the problem of differentiation noise in the velocity signal, which affects control and perception of force-feedback audio-haptic interaction. In the literature this problem is often resolved simply by application of a low-pass filter, or if the benefits of a new estimator are examined, it is often considered only from a stability point of view. Most often the virtual wall impedance test as described by Colgate and Brown [31] is given as evidence of improved performance in terms of maximum stiffness, but the perceptual detection of noise and delay is usually ignored unless it causes exceptionally obvious problems. Moreover the friction case is often considered secondary to wall damping, despite being an important type of interaction particularly for musical scenarios.

Partly this may be because operators expect a certain amount of imperfection in haptic display, due to mechanical friction for example, and are willing to ignore subtle noisiness. We found, in fact, that operators are willing to ignore a surprising amount of noise during interaction, to the point where we had difficulty having subjects identify it until we gave them direct control over the noise level. However, this argument does not hold for audio applications where high-quality output is the *raison d'être*.

Since we are interested in high-fidelity musical interaction, the appearance of noise in the audio signal was a clear issue, however the trade-off for delay imposed by filtering can also be problematic. Therefore we examined several methods previously proposed for improved estimation, and also explored the use of velocity and acceleration sensing, comparing the performance of each for our intended scenario using both off-line numerical analysis and on-line evaluation.

9.2 Contributions

From the results, we may make some recommendations for the design of force-feedback interfaces in terms of sensing apparatus, and we can also suggest the utility of a pre-processing stage in the digital servo pipeline for input signals.

9.2.1 Recommendations for hardware design

Firstly, it is clear from Chapter 7 that there is an advantage to direct velocity measurement by means of an analog tachometer. It allowed for a very strong increase of the impedance

range of bowed-string friction display, effectively doubling it compared to a low-pass filter, with very low delay in velocity detection and by far the lowest noise level compared to any estimator, a difference of between one to two magnitudes. Therefore we would recommend this technology as an interesting candidate for integration into future haptic devices.

The sensor we used for this study, the Trans-Tek LVT, has a large form factor, but it is similar in shape to an LVDT linear position sensor and could be attached in serial or parallel. Like the LVDT, however, it does add significant friction to the mechanism; it would be interesting to design lower-friction sensors of a similar nature.

On the other hand, considering that a tachometer is effectively an inverted motor, it is also possible to sense velocity from the motor's back-EMF signal, as used e.g. by Weir et al. [132] for active electrical damping. This could be a useful input for the digital portion of the servo loop, averting the need for extra sensing apparatus. Conversely, some methods suggested in our work may be interesting to apply to digital control of active damping circuits, which may be executed independently from the main simulation loop, possibly at a higher rate.

Accelerometers could yet be shown to have benefits for haptic devices. In the current work, we were unable to clearly show improvements using acceleration measurement as we had initially intended. On the other hand, the optimiser's rejection of accelerometer data, Chapter 6, may have been a consequence of our strategy. It is possible that allowing some amount of noise in exchange for reduced delay may have enabled the emphasis of position-based noise in preference to a less noisy but biased accelerometer integration, therefore further research into parameter tuning strategies may provide more insight.

9.2.2 Signal processing

We found that some estimation techniques did simultaneously outperform others on both criteria of noise and delay. In particular the combination of the 2-sliding Levant observer with the Kalman filter managed to give impressive noise results while being very competitive on our delay criterion. Meanwhile the hybrid methods did not perform best for our online impedance range measures, but also did not by and large perform worse than previously-proposed methods, while outperforming them on noise. To our knowledge this is a novel result. Investigation of the disagreement between our delay evaluation and performance in the online maximum impedance test is on-going.

Since several methods did prove taxing on the processor, forcing us to lower the sampling rate, it suggests that these might be interesting candidates for integration into digital circuitry in the servo, or to be implemented on a dedicated signal processor, in order to alleviate the core processor.

We did not experiment with execution of the estimation algorithm at faster rates than the simulation, however several methods such as Levant's differentiator can be expected to increase in quality as higher sampling rates are used. Using digital logic with a fast and high-resolution analog-to-digital converter may allow significant improvements to estimation.

9.2.3 Haptic simulation

In haptic simulation, we found in Chapter 3 that, in the presence of measurement noise, virtual coupling methods based on dampers or critically damped springs should transmit less noise to the model than position-only spring couplings.

However, we also showed that virtual coupling for the purpose of inertia simulation may be successful for haptic simulation, but may not properly simulate the desired inertia at higher frequencies. Nonetheless, in bowed string simulation, a high-stiffness spring representing the bow hair is more likely to affect acoustic response than the device-bow coupling, since mass-spring systems have strong low-pass characteristics.

Finally, we found in Chapter 4 that it was possible to remove noise by driving the acoustic model based on impulses determined by a position-based friction model.

9.3 Future research

The success with the Kalman filter combined with fast estimators suggests that we may be able to improve results further by feeding the Kalman filter with even more measurements and estimators. We showed some attempts to use acceleration measurement, but we did not for instance try using the tachometer as a Kalman filter input.

Additionally the Kalman filter was kept extremely simple in this work. It was provided only a double-integrator model, the relationship between position and its derivatives. It may perform much better given a complete model of the device. With the addition of inertia and friction to the Kalman process model, of course using non-linear Kalman methods for the latter, it may also be interesting to use force sensing in place of or in addition to acceleration.

Of course, most haptic devices use rotary mechanisms rather than the linear system used in the TGR, and often make use of optical encoders for position sensing, which may be used with different velocity sensing techniques. Evaluating the current methods in comparison with encoder-specific techniques may be needed to complete this survey of differentiation methods. Rotatory mechanisms also allow for other types of sensors: a dynamo can be used as a tachometer, effectively using the same theory as the LVT, and similarly a MEMS gyroscope may be interesting to evaluate since it measures angular velocity.

In regards to optimisation, we used an objective-sum method because we valued both error and delay properties of the estimators, and wished to compare the “best” configuration of each on both the criteria. However, it may be more clear to compare only a single variable at a time, and therefore other approaches may be appropriate. For example, *bounded objective function* methods may provide the possibility to establish an inequality constraint on error or delay, while optimising the opposite variable [77]. Thus, all estimators would be considered to have equivalent delay, and then we would compare only their performance on error, or vice-versa. This approach was in fact followed initially in this work but abandoned since not all estimators were able to achieve the desired delay constraint. However, more work on this topic is necessary, and it may be appropriate for example to simply reject estimators that do not satisfy the inequalities.

We wonder whether such an approach might allow the optimiser to make better use of accelerometer information, for instance to see whether the acceleration data helps improve delay within some error bound. In theory acceleration measurement should provide faster estimation of velocity due to direct measurement of its derivative, which was partially confirmed by the fact that we had most success with acceleration measurement in high-gain conditions. Therefore, further investigation of the applicability of accelerometers to haptics must be conducted.

Nonetheless, it would be nice if the parameters were more robust to variations in signal quality, since additional noise or a misestimated offset or drift in a sampled signal can cause the results to vary widely. Although this should not happen, in practice it is a question of robustness, and may be a concern for commercial applications. Therefore it may be interesting to develop on-line optimisation methods which should provide much more reliable performance; however it is not clear how to proceed since the optimisation performed in this work relied on removing delay using cross-correlation, a strictly off-line procedure. However, if delay and noise can both be estimated in an on-line manner, perhaps

using a sliding window approach, it may be possible to develop such a technique.

Moreover, we would like to make similar investigations at higher sampling rates. Initial work on this project was performed using 20 kHz signals, but we chose to work at 5 kHz when it was realized that we could not run several algorithms concurrently at that rate for testing purposes. With lean estimators, we are able to execute the model at 35 kHz, which produces much higher-quality audio. Of course, in practice much of this is filtered by the inertia of the device, but certain estimators such as Levant's differentiator should improve in precision and accuracy with frequency [28], and therefore it would be interesting to see how the performance of each estimator scales with frequency.

In addition to technical development, as we have mentioned there would be some benefit to future study of this problem from a purely perceptual point of view. It is not clear how best to design psychometrics that may determine the quality of a haptic simulation in the presence of noisy signals, since the results depend heavily on the interaction between the haptic interface, the model, and the user's preference. Therefore some attention to design of quality measurement methods is needed for the design and evaluation of audio-haptic devices and simulations. In particular we wish to develop methods for ensuring that distortion due to signal noise and delay is below the level of human perception, ideally parameterized according to device specifications and model coupling design.

To prove the generalizability of methods proposed in this work, we would like to implement interactions with other models, especially musical interactions that more naturally involve higher friction forces. Interactions with other objects, such as bars, tubes, glass, and textured surfaces deserve some attention, and could be useful for musical and foley purposes.

As for the bowed string, we would certainly like to apply these techniques to multi-DOF devices in order to allow high-fidelity control over the many axes involved in bowing. This would allow string selection as well as control over skew, roll, and distance from the bridge, all parameters which may affect not only the sound but also the gestural dynamics of bowing [110]. For multi-DOF devices with complex kinematics, the use of accelerometers may require preprocessing in order to normalize coordinate systems and eliminate gravity bias, however the addition of an accelerometer at the end effector may actually help to reduce lag in both position and velocity estimation due to non-ideal stiffness of linkages. For example, this can help with high-frequency acceleration matching to ensure that desired vibrations are reproduced accurately [81]. If an accelerometer is to be used for HFAM,

the additional benefit of contributing to the velocity and position estimation is a natural extension of such a proposal.

Finally, there exist other uses of improved velocity estimation with low delay in both haptics and other areas that would be worthwhile to explore. The use of a passivity controller [52] provides a guarantee of stability for arbitrary force rendering by tracking the excess energy in a system, however its performance is limited by noise introduced in the dissipation forces since they are velocity-dependent. Another use for high-quality velocity information is in friction compensation for smooth control of servo mechanisms [7].

9.4 Concluding remarks

While there is still much work to be done in developing and fully evaluating some ideas related to this work, we hope that on the whole the findings presented here will be taken as supporting evidence for the importance of high quality input signals in the fidelity of force feedback simulation. Our goal was to understand more fully the trade-offs involved in noise elimination, and if possible, to reduce them.

If virtual environments are ever to be used in music and media arts in a serious way, we expect that, apart from affordance issues such as workspace and degrees of freedom, quality of perception will be the deciding factor. More so, even, than characteristics such as maximum stiffness and peak force, we tend to side with the supposition that subtlety and precision will be the critical factors in perceptually-convincing rendering. Therefore, we hope that this work may lead to further study of perceptual issues in high-fidelity, high-rate haptic display and interaction with media and music.

Appendix A

Equipment

In this appendix we give details on the equipment used throughout this work for force feedback implementation and testing.

A.1 The Ergon-X system

Throughout this work we made use of the Ergon-X system from ERGOS Technologies, also known as the *transducteur gestuel rétroactif*, or TGR, mentioned in Chapter 1 as designed by the ACROE laboratory at INPG, France. This equipment was made available through collaboration and funding by a Special Research Opportunity from the Natural Sciences and Engineering Research Council, in connection with the Enactive European Network of Excellence.

We will use the term TGR to refer to the device mechanism, consisting of a set of linear vertical “slice” motors, each coupled to a Linear Variable Differential Transformer (LVDT) displacement sensor. The Ergon-X system additionally consists of a linear current amplifier for driving the motors, and a real-time signal processing board called *Toro-16* from Innovative Integration Inc.

This DSP card, featuring a Texas Instruments 320C6711 float-point signal processor, can perform 16-bit analog-to-digital and digital-to-analog conversion at variable sampling rates up to approximately 250 kHz, depending on the demands of in-loop computations. Typically algorithms executed between 5 and 40 kHz.

This exceeds the usual range for haptic computation, which allowed us to explore the problems and possibilities of high-rate force feedback.

<i>Property</i>	<i>Value</i>
DOF	Configurable
Max. peak force	200 N
Max. continuous force	60 N
Max. stiffness	40 N/mm
Peak acceleration	60 m/s ²
Resolution (basic)	1 μ m
Position noise* (basic)	14 μ m
Friction (max. force)	5×10^{-3} N
Weight*	300 g / slice
Workspace basic*	2.2 cm
Workspace key	5.0 cm
Workspace 2-DOF	15 \times 15 cm
Workspace 3-DOF	6 \times 6 \times 2.5 cm
Power	1 mW max.

Table A.1 Properties of the Ergon-X system, from data sheet. Those marked with * have been confirmed or determined by measurement.

Device properties are listed in Table A.1.

A.2 Calibration

The software on the Toro-16 card sees the input and output as two arrays of floating point values in the range $[-1, 1]$. The DSP runs a main routine which initializes the algorithm parameters, sets up an interrupt handler, and then waits for exit instructions. The interrupt handler is called on every clock cycle, and its job is to read the A/D registers, compute the force command, and write it to the D/A registers.

In order to work in physical units, we calibrated the device by means of manual measurement using electronic measuring equipment. A set of electronic calipers was used to measure the displacement range for a motor, which came to 2.2 cm, with the center of the range reading 0. Therefore a coefficient of 0.0245 was used to convert to meters, since a few millimeters on either side of the numerical range are saturated in the LVDT range.

To calibrate force output, an electronic kitchen scale was used which featured a pair of wires that provided the measured force as a pulse-width signal. The device was turned upside-down, placed on blocks, and the motor was extended with a rod in order to push



Fig. A.1 The TGR device in 2-DOF configuration with a bow attached. We did not use this configuration for testing, however a simple, not-very-robust coupling using tape was attempted briefly to explore how a bow would feel when attached to the device.

down on the scale. Several heavy books were placed on top of the machine to ensure that it would not exert enough force to move, and data acquisition hardware was connected to the scale's two-wire output. The scale was zeroed to remove the initial weight of the motor. A series of DC forces were commanded, and the weight in grams was recorded. After several samples, the resulting recording was fitted to a linear curve, and converted to force in Newtons. A final coefficient of 188.47 was found by this means.

The force calibration was also repeated using a manual spring scale, and a similar coefficient was found.

Note that the device amplifiers and LVDT driver feature adjustable analog offset knobs for both position and force, allowing to center the range according to the physical adjustment of the LVDT rods. This adjustment step is performed using a method developed at ACROE, where a spring is displayed on each motor, and each one is adjusted such that the



Fig. A.2 The device calibration setup. The device is upside-down balanced on two blocks, with clamps used to ensure that the blocks do not tip. A rod attached to the motor pushed on an electronic kitchen scale. Heavy books are piled on top to ensure the device, which was not clamped to the desk itself, does not move upwards when exerted downward pressure on the scale.

motor is centered and force offset is adjusted to compensate for gravity. In some cases we used the device in a side-ways configuration, and in that case we adjusted the force offset such that the motor would not move when the force command was zero—in other words, the force offset was adjusted to a value below the device’s force of static friction.

Note also that the friction and inertia of the device were modified by attaching the LVT sensor, described in Chapter 5. This difference is characterised in the next section.

A.3 Friction characteristics

The tachometer was attached to the device via a bar linkage connected in series with the LVDT bar through a mechanical coupler made of aluminum. When connected, the friction

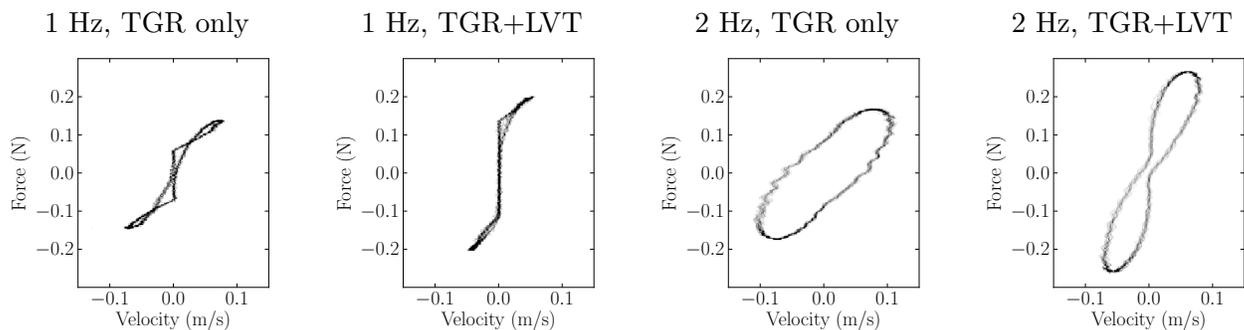


Fig. A.3 Force-velocity plots of sinusoidal signals showing friction behaviour with and without LVT tachometer sensor attached. Force sinusoids of 1 Hz were used at a low amplitude, while recording force and velocity measurements. In both cases, an FOAW *best-fit* velocity estimator was used with a window size of 32 points to estimate the velocity from the position sensor. It can be seen that at low velocities the static friction was not negligible.

of the LVT sliding in its coils added non-negligible friction to the interaction.

Although we did not actively compensate for this, we used parameter identification to characterise the difference in friction. We identified the friction parameters of the device with and without the tachometer attached using methodology outlined in [105].

This methodology proposes to use a position sensor for velocity estimation, a force sensor and an accelerometer to inform a least squares approach to friction model fitting. A sinusoidal signal may be used to move the system at a range of low velocities as measurements are recorded.

We did not use a force sensor, but rather recorded commanded motor force. Since we needed to compare conditions with and without the tachometer, we recorded position data and used an FOAW *best-fit* velocity estimator with a 32-point window. (See Chapter 5.) We also used an ADXL-203 accelerometer from Analog Devices configured at a cut-off frequency of 2.5 kHz. Due to equipment availability we were unable to use the PCB Piezotronics accelerometer for this task.

Recordings of sinusoids at 1 and 2 Hz as force-velocity plots can be found in Fig. A.3. It can be seen that the change in friction for the tachometer is non-negligible, modifying the response at low velocities quite significantly.

Using the fitting method from [105], we performed identification for parameters of the Karnopp friction model [63]. These include mass, as well as static and dynamic friction

<i>Method</i>	<i>Frequency</i>	<i>LVT attached</i>	<i>Mass</i>	C^+	C^-	b^+	b^-
Standard	1 Hz	No	0.0403	0.00843	0.0170	1.92	1.94
Standard	1 Hz	Yes	0.120	0.0290	0.0238	4.15	4.60
Standard	2 Hz	No	0.305	0.0282	0.0364	0.969	0.942
Standard	2 Hz	Yes	0.376	0.0842	0.0685	2.03	2.28
Covariance	1 Hz	No	0.0487	0.00735	0.0157	1.95	1.98
Covariance	1 Hz	Yes	0.172	0.0274	0.0217	4.24	4.70
Covariance	2 Hz	No	0.603	0.0259	0.0339	1.00	0.980
Covariance	2 Hz	Yes	0.594	0.0826	0.0666	2.08	2.33

Table A.2 Friction characteristics as reported by least squares fit. C^+ and C^- are the positive and negative static friction coefficients, while b^+ and b^- are the positive and negative viscous coefficients, respectively.

coefficients for positive and negative velocities. These terms are collected in a matrix β , such that the relation,

$$\begin{bmatrix} F_1 \\ \vdots \\ F_n \end{bmatrix} = \begin{bmatrix} a_1 & \text{sgn } v_1^+ & v_1^+ & \text{sgn } v_1^- & v_1^- \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_n & \text{sgn } v_n^+ & v_n^+ & \text{sgn } v_n^- & v_n^- \end{bmatrix} \begin{bmatrix} m \\ C^+ \\ b^+ \\ C^- \\ b^- \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix} \quad (\text{A.1})$$

$$F = \mathbf{X}\beta + \epsilon \quad (\text{A.2})$$

reflects the equations of motion for the Karnopp model. Here, \mathbf{X} and F are recorded measurements and ϵ represents the measurement error. The division into position and negative components allows for standard linear least squares methods:

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} (\mathbf{x}^T F) \quad (\text{A.3})$$

Additionally, [105] suggests the following modification, which allows for the expected measurement covariance to be specified in a matrix $\ddot{S}_{uu} = \text{diag}(s_a, 0, s_b, 0)$, where s_a and s_v are the acceleration and velocity covariance respectively:

$$\hat{\beta} = (N^{-1} \mathbf{X}^T \mathbf{X} - \ddot{S}_{uu})^{-1} (N^{-1} \mathbf{x}^T F) \quad (\text{A.4})$$

We tried both methods. The results, listed in Table A.2, were not perfect, but give some information on the expected friction qualities for the device with and without the tachometer. The main issue is that the algorithm did not correctly or consistently report the mass. Fortunately, there is more consistency in the reported friction and viscosity coefficients, and these follow the expected trend. It is clear that there is an increase in friction due to the tachometer between about 1.5 and 3 times.

Since we were not compensating for friction, this exercise was adequate for us to get an idea about the friction qualities of the device.

Appendix B

Parameter tuning

The algorithms under investigation in Chapter 6 were tuned by automatic global multi-object optimisation. However, to understand and contextualize these parameter choices according to our problem set, it is beneficial to visualize the objective function surfaces for each algorithm.

This appendix contains such visualisations and some discussion on how the final choices apply in the context of each individual recording we examined.

B.1 Automatic optimisation challenges

Leading up to this work, we performed individual numerical optimisation on each estimator using a brute-force global optimiser. Unfortunately we found this approach to be inconclusive, since for some estimators, the global optima tended to recommend a wide range of parameter choices that changed depending on the exact recording used. These inconsistencies made it difficult to combine the individual results into a globally optimal parameter, as well as creating difficulty in comparing between estimators.

A better approach was evidently the multi-objective optimisation discussed in Chapter 6. However, we nonetheless sought to understand why the individual optimisation strategy failed. In investigating the reasons for the observed behaviour, visualization of the error surface revealed that not all conditions contained convex valleys with sharp and well-defined bottoms, therefore the exact choice by the optimiser could change quite drastically due to differences in the input. Equally challenging for interpretation, we noticed that for some estimators, there existed global optima that were unique to particular record-

ings, while across several recordings there nonetheless existed a local minimum that was more consistent between cases; the individual optimizer would never make the more reliable choice.

Although these problems were surmounted more or less by the use of a multi-objective approach, effectively by summing the normalized objective spaces for each individual condition, nonetheless these visualizations provide some useful insight into the behaviour of each algorithm.

Thus, in the sections that follow, the error and delay surfaces of each estimator will be provided based on an exhaustive grid-based enumeration of its parameter space.

B.2 Parameter selection with respect to error and delay surface inspection

Here, we give the objective function surfaces for each estimator, with discussion.

B.3 Position-based estimators

We first look at the behaviours of purely position-based estimators: the simple low-pass filter; the least squares FIR filter; the first-order adaptive windowing filter (FOAW); Levant's 2-sliding observer differentiator; finally, the 2nd-order Kalman filter.

B.3.1 Low-pass filter

The error surface of a 2nd-order Butterworth low-pass filter applied to the position differential can be seen in Fig. B.1.

As expected for a low-pass filter, the error decreases from a highly-distorted signal at low cut-off, an optimum is found, and as fewer and fewer high-frequencies are blocked, the noise due to differentiation begins to drastically affect the signal quality. It is clear that there is a valley present between approximately 20 and 200 Hz.

However, a more precise choice is not obvious. The bottom of this valley is fairly flat for most recordings. At low friction, the preference between the upper and lower range of the valley is not clear, as some examples have a minimum at 20 Hz while others are nearly

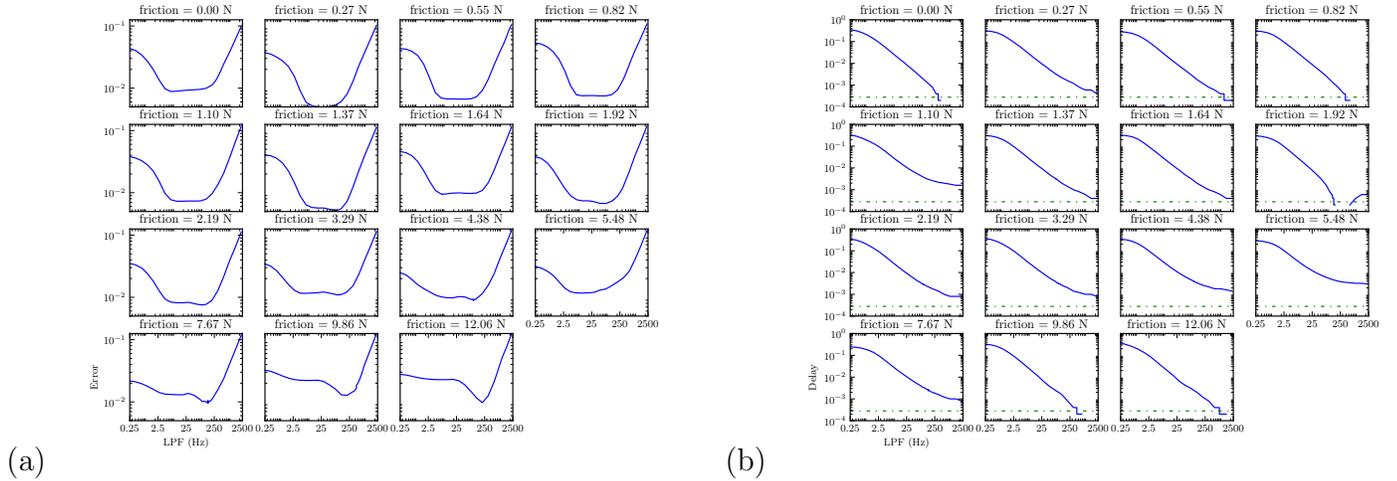


Fig. B.1 Error and delay surfaces for a 2nd-order Butterworth low-pass filter applied to each recording. The valley has a flat bottom for low friction conditions, but increases sharply in the higher frequencies as the friction is increased. Meanwhile, the delay decreases log-linearly as the filter cut-off increases.

completely flat. At high friction, however, the preference for 200 Hz to 250 Hz becomes much clearer.

Correspondingly, the delay decreases log-linearly as the cut-off frequency increases. If we consider delay, then a preference for higher cut-off frequencies is clear.

B.3.2 Least-squares FIR filter

It can be seen in Fig. B.2 that the error characteristics of the least squares estimator are extremely consistent across examples. However, the delay behaviour is much more difficult to interpret.

It is notable that the delay behaviour is quite different between 1st-order vs. 2nd- or 4th-order least squares. In the former, it generally increases log-linearly with the window size, with some exceptions. However, the latter exhibits very sharp changes in delay, with a clear bottom between about 5- to 20-sample windows. For the 4th-order fit, the walls of this valley are extremely sharp, indicating a potentially disastrous result if too-small or too-large windows are used.

In the error surface, it would appear that the 1st-order least squares fit is actually the

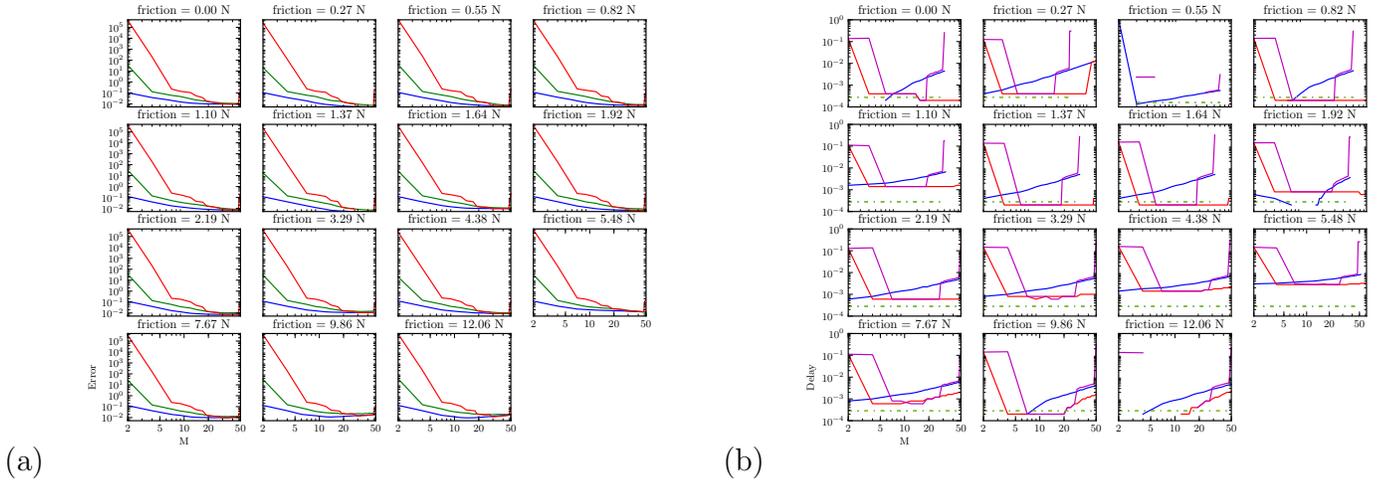


Fig. B.2 Error and delay surfaces for the least squares fit FIR estimator applied to each recording.

best, however it also seems to have the worst delay characteristics; the slopes of error and delay for the 1st-order choice are inversely proportional (on a log scale). Therefore it would seem that 2nd- or 4th-order choices are better if delay is important.

Since larger windows require longer computation time, we would like to choose a relatively short window. For both error and delay, it is clear that a 2nd-order fit affords better results for smaller windows, therefore this seems to be the best choice.

We also notice that a window size around 20 to 25 converges to approximately the same error characteristics as the 1st-order fit, while having better delay properties. For high friction, a smaller window would be a better choice in terms of delay, but at the expense of several magnitudes more error.

Note that it appears in Fig. B.2(a) that the error never seems to increase, such that there is no valley optimum. However this is somewhat misleading. In the last plot (friction = 12.06 N), the error curve begins to increase slightly at approximately $M=40$, though it is not very apparent visually. Unfortunately we did not test longer window sizes, as they would be prohibitive for online computation.

B.3.3 First-order adaptive window best-fit

Since the FOAW algorithm has two parameters, position noise margin and maximum window size, we give an example of the full 2-D error surface in B.3. However, it is immediately

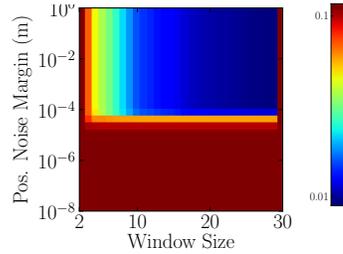


Fig. B.3 A two-dimensional error surface for one recording of the FOAW best-fit estimator demonstrating the separability of the parameters. Below a threshold of approximately 10^{-4} m, the performance is bad, but above this threshold the error depends only on the window size. This pattern holds for all recordings, therefore we give one example here.

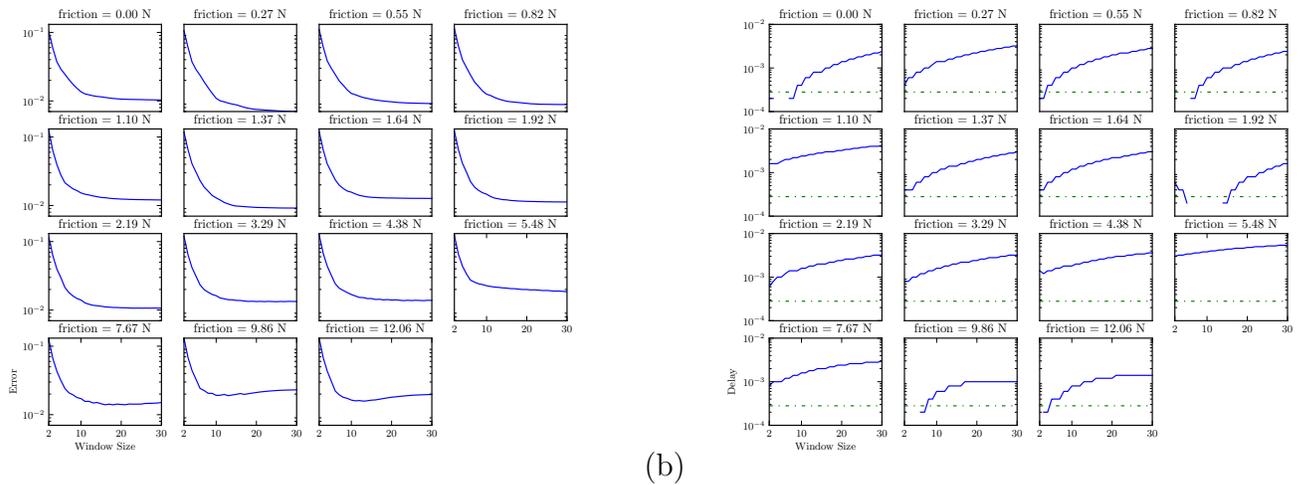


Fig. B.4 The 1-D cross-section of the FOAW error surface along the window size axis at a position just above 10^{-4} m for (a) error and (b) delay.

noticable that the two axes are independent and that a particular choice for the noise margin is obvious: There is a threshold for the noise margin between 10^{-5} and 10^{-4} m, below which the error is maximized.

Above this margin the error behaviour seems to depend only on the window size. Therefore, in Fig. B.4, we take a 1-D cross-section of the error and delay surfaces along the more interesting window-size axis at a grid location just above this threshold, about 7.2×10^{-5} . We note that the threshold, although a few times higher, is on the same order as our measured position noise amplitude of $14 \mu\text{m}$, mentioned in Chapter 5. This is expected since the noise margin, determined by the signal amplitude when the device is at rest and the position is at zero, is higher than the corresponding RMS.

Turning now to Fig. B.4(a), the decaying nature of the window-size axis shows that the FOAW algorithm tends to get better with longer windows, except at high gains, where error starts to increase past about 10- or 12-point windows. This reflects its first-order fit, used in the FOAW *best-fit* method, which should give a more accurate result for low velocities. On the other hand, we would expect it not to be punished on error for long choices of the maximum window size, due to its ability to decrease the window size as necessary during on-line operation. However, we noted in Chapter 6 that this was not, in general, the case—most of the time, the window size was maximized, incurring delay.

However, it seems, from the two highest gains at 9.86 and 12.06 N, that the delay tends to hit a maximum at around 20 points, which is also where they begin to incur more error. This suggests that for high-gain conditions, the window size is indeed being dynamically shortened.

Although this ability to adapt the window size is beneficial for balancing noise rejection and time response, this behaviour actually presents difficulty for automatic optimisation, since the window size limit may be increased arbitrarily, but it does not imply that the full window will actually be used. Therefore we should not expect an unlimited decrease in quality as window size is increased, yet longer windows may afford exponentially smaller and smaller improvements as more data is averaged during low-frequency periods. An automatic optimiser may thus tend to increase the maximum window size arbitrarily and continue to see slight improvements; meanwhile, the returns for occasionally performing a large averaging operation are smaller and smaller, adding unnecessarily to the processing cost. This makes clear that processing costs could also be a useful objective to include in an optimisation procedure.

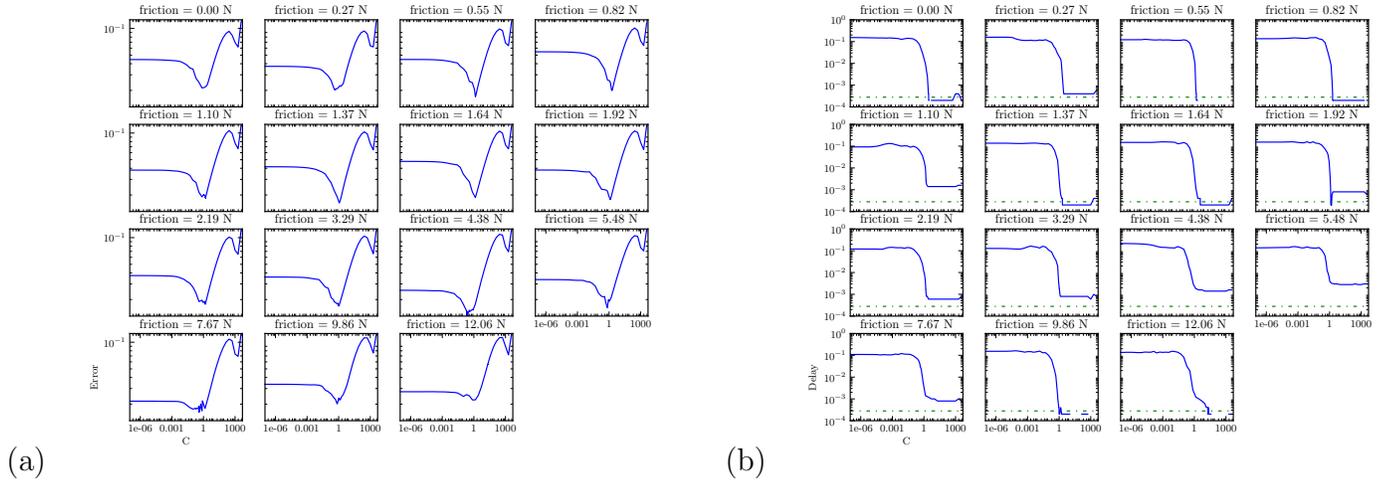


Fig. B.5 (a) Error and (b) delay surfaces for Levant’s 2-sliding observer differentiator applied to each recording. A clear optimum is present in the error, while delay reduces quite immediately when the single parameter, C , is correctly tuned.

B.3.4 Levant’s differentiator

The Levant 2-sliding observer differentiator features only a single parameter, C , which is a Lipschitz constant representing the maximum expected acceleration. We used a very wide range for testing from 10^{-7} up to 5000 since we preferred to make no assumptions and observe how the error response might converge.

This is one case where there is a clear optimal choice for the parameter, since for every recording, (Fig. B.5,) there exists a highly emphasized notch at a particular location, almost exactly $C = 1$. Below this point, the system can easily diverge from the followed signal, whereas above it, switching noise will cause the result to oscillate around the true value with increasing amplitude.

We can see from Fig. B.5(a) that this switching noise indeed increases quite dramatically after the optimum. Similarly, Fig. B.5(b) shows that the delay diminishes quite rapidly once C is adequately tuned, and stays low, even while noise increases.

It is not known why the error tends to descend slightly for very high C , though, it could be related to the specific structure of the bowed-string signals.

It is recommended by Levant [72] that the differentiator be followed by a filter in order to remove the switching noise. We investigated this possibility by post-filtering using a

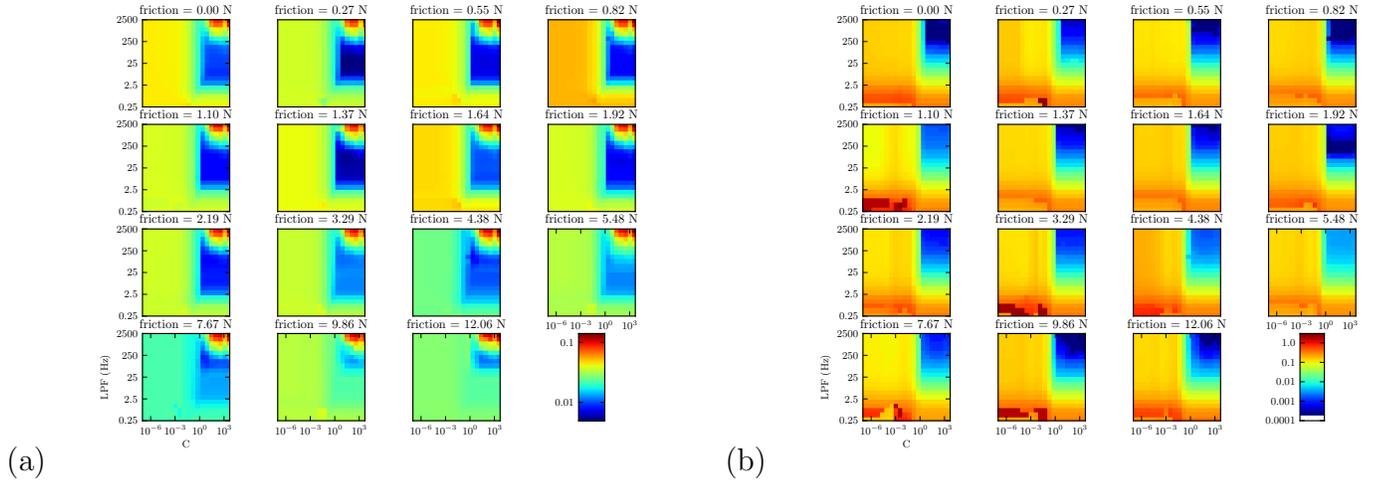


Fig. B.6 (a) Error and (b) delay surfaces for Levant's 2-sliding observer differentiator, post-filtered with a BW2 low-pass filter, applied to each recording.

2nd-order Butterworth low-pass filter, giving a second parameter to tune. The results are given as 2-D surfaces in Fig. B.6. It can be seen that, at least for low friction gains, the low-pass filter helps to remove the sharpness of the optimal notch, between frequencies of 2.5 to 250 hz. At the same time, good delay behaviour is found for frequencies above about 50 Hz.

There is some amount of curvature in the upper-right region where the surface begins to increase rapidly, Fig. B.6(a), indicating a dependence between the two parameters. One can infer that there is something to be gained by keeping C close to its lowest value, since higher filter cut-off frequencies can be used, decreasing the amount of delay incurred. Close to $C = 1$, it seems possible to set the frequency to 500 Hz or even slightly higher, without increasing the error significantly.

At higher friction gains, however, the optimal area is much smaller, and the only choice of parameters remain around 200 Hz with $C \approx 100$. Not coincidentally, this is the same choice as for our simple low-pass filter differentiator.

Although $C = 1$ is probably a good choice, in practice we did find we needed a somewhat higher value, around $C = 10$ or so, in order to ensure that the observer did not temporarily diverge—in particular, these evaluations look only at the recorded gestures, but do not consider for example the high accelerations experienced when the device is tapped against

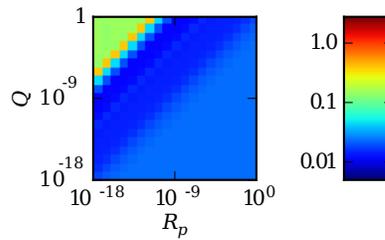


Fig. B.7 Example of the 2-D error surface for the 2nd-order Kalman estimator. It can be seen that there is a strong inter-dependence between the two covariance settings for the process (Q), and measurement (R_p). Therefore, for analysis we instead examine the diagonal cross-section of this graph in Fig. B.8.

the workspace boundary.

B.3.5 2nd-order Kalman filter

Similar to the case of the FOAW results, for the Kalman estimator we examine a 1-D cross-section of the error surface. In the FOAW analysis, we saw that the effect of each parameter was quite independent, and therefore could be examined separately. In the case of the Kalman estimator, the opposite is true: from the 2-D surface presented in Fig. B.7, we can see that the two covariance parameters, that of the process Q , and that of the measurement, R_p , are completely inter-dependent. That is, it is not their absolute values that matter, but rather their ratio that determines the behaviour of the Kalman observer.

This is intuitive if we consider that the role of the covariance parameters is to help determine the Kalman gain K which balances the mixing of these two measurement sources. Whatever their values, K must have a combined total of 1 in order to reproduce the correct measurement. Thus it makes more sense to examine the diagonal cross-section of the 2-D surface, such that the horizontal axis represents the ratio between the two parameters. This is presented in Fig. B.8.

From Fig. B.8(a), the optimum is unique and steady at a log-ratio approximately $\log R_p / \log Q = 2$. This nicely corresponds with the delay optimum, Fig. B.8(b), which begins to increase dramatically at a log-ratio below 2.

The behaviour below this ratio tends to produce low-pass-filtered, slow responses, while above this ratio the output is able to better-follow high-frequency changes in the input signal, however it is at the cost of increased overall error. The nominal best performance

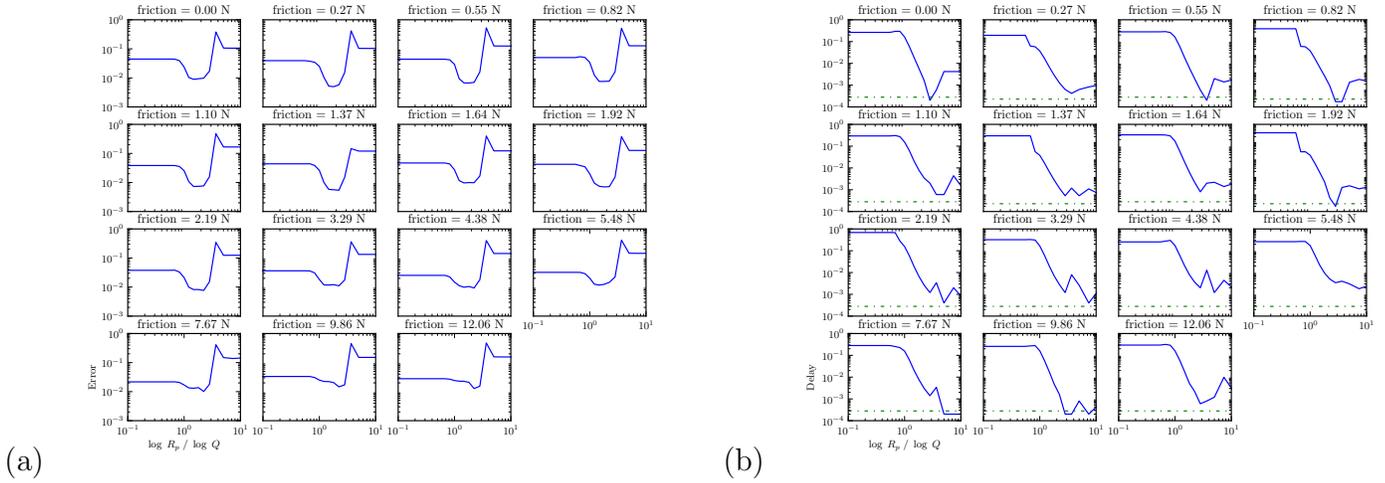


Fig. B.8 One-dimensional (a) error and (b) delay surfaces for the 2nd-order Kalman estimator applied to each recording, actually the diagonal cross-section of Fig. B.7; thus, the horizontal axis is the ratio between process covariance Q and measurement covariance R_p . An error optimum is clear, and coincides well with the delay optimum.

at a ratio of 2 tends to follow the input velocity low-frequency shape, while flattening out the high-frequency stick-slip behaviour.

Our position measurement error was measured at an RMS amplitude of 14 μm , or a covariance of $2 \times 10^{-10} \text{ m}^2$. For $R_p = 2 \times 10^{-10}$, a log-ratio of 2 corresponds to $Q = 1.4e - 5$.

B.4 Fusion estimators incorporating accelerometer measurement

The previous section discussed estimators that made use solely of the LVDT position sensor, yet we have conjectured that velocity estimates can be improved by incorporating accelerometer measurements.

Here, we discuss the parameters for algorithms that use both sensors. These have higher-dimensional parameter spaces since there is generally at least one parameter per sensor.

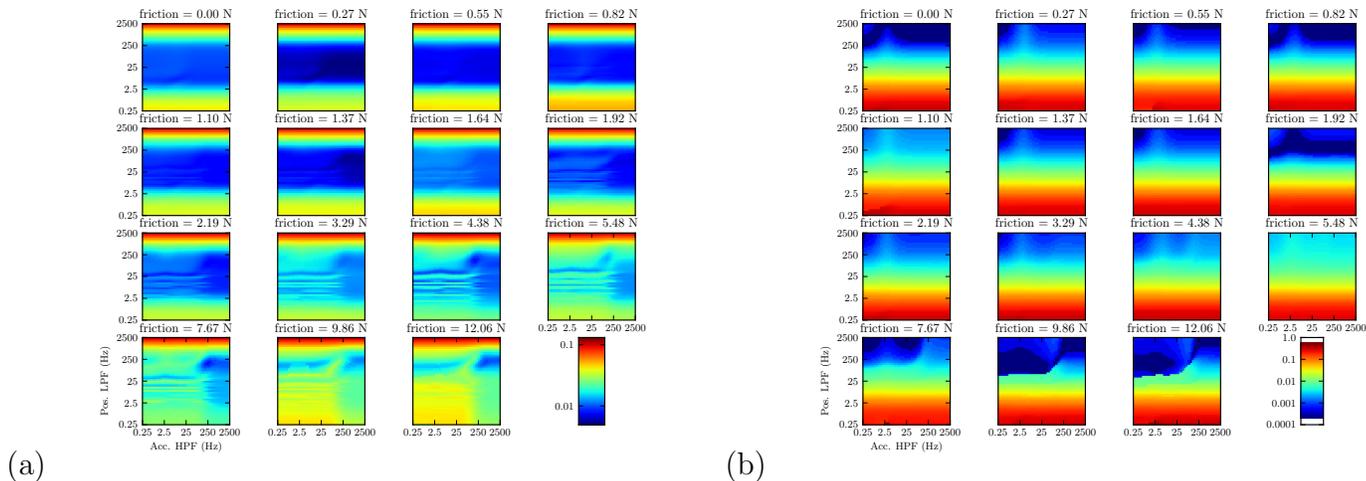


Fig. B.9 (a) Error and (b) delay surfaces for the complementary filter estimator with varying parameters for BW2 low- and high-pass filter cut-off frequencies. It can be seen that the accelerometer contribution evolves significantly as friction gain is increased.

B.4.1 Complementary filter

The complementary filter linearly combines position and acceleration measurements by mixing frequency bands in which each sensor performs best. By using independent filters for each band we could examine how the total response performs with respect to this two-dimensional parameter space. The diagonal cross-section of this space of course corresponds to the performance in a strict complementary configuration with a single cross-over frequency, however we found it informative to visualize the full two-dimensional sweep of each independent frequency, to help shed light on our difficulty in using automatic optimisation for this estimator.

From Fig. B.9(a), we can see that as friction gain is increased, the contribution of the accelerometer evolves in a particular, non-convex fashion. At low gain, the accelerometer effectively makes no difference, while the position cut-off adjustment behaves as the low-pass filter described in section B.3.1. However, at high gain, the contribution of the accelerometer develops a striped pattern with a series of minima along the position axis for low values of the high-pass filter frequency. For higher values of the HPF, this striped pattern disappears at moderate gain, but at very high gain it persists.

Even using global optimisation it was difficult to consistently identify optimal param-

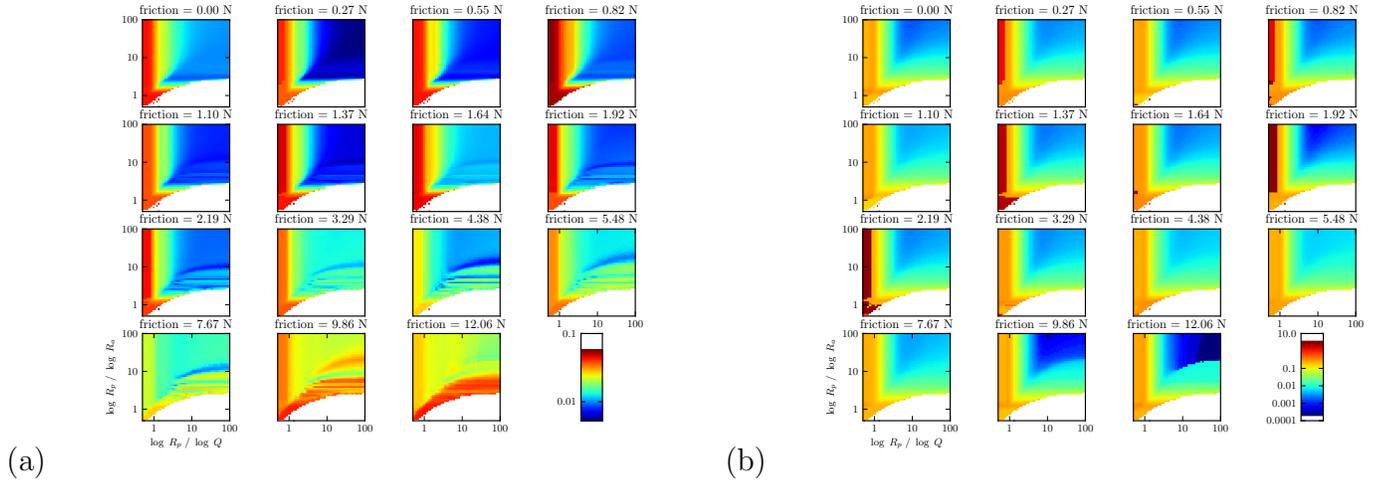


Fig. B.10 Two-dimensional view of the (a) error and (b) delay surfaces for the 3rd-order Kalman estimator with varying parameters for process Q and accelerometer R_a covariance, expressed as log-ratios of R_p . Position covariance R_p is held constant at $2 \times 10^{-10} \text{ m}^2$. White regions in the lower-right corner indicate parameters that lead to divergence.

eters: for low gain, the flat bottom caused accelerometer choices to vary wildly; for high gain, these striped areas frequently featured the lowest minima, even though these minima varied significantly between recordings. The 2-D visualisation in Fig. B.9(a) allows us to identify a clear point of stability in the parameter space at approximately 250 Hz on both axes; despite the fact that this location was not consistently the global minimum, it is nonetheless a good choice due to its consistency across gain settings. It is also close to the area of low delay, Fig. B.9, which depends foremost on the position filter.

The exact reason for the striped shape of the error space along the position axis is not known, however we believe it to be related to an interaction between filter delays and the frequency of the string oscillations. However, more empirical investigation would be needed to confirm this.

B.4.2 3rd-order Kalman filter

The 3rd-order Kalman filter mixes both position and acceleration measurements with a process model estimate. Although nominally this is a 3-dimensional parameter space, with covariances for two measurements and one for the process, we can consider, similar to the

2nd-order case, that it is the log-ratios between covariances that are important. Between 3 covariances, therefore, we have a 2-dimensional log-ratio space which we can visualise as a 2-D surface. Choosing the same absolute value for $R_p = 2 \times 10^{-10} \text{ m}^2$, this parameter is held constant while R_a and Q are varied to give the error and delay surfaces shown in Fig. B.10.

We can see that the resulting surfaces have a descending shape towards higher values of both variables. Interestingly, as both log-ratios are increased, there is no tendency for the error or delay to begin increasing again. We verified by extending the range that this pattern does not change for very large log-ratios, but rather the surface flattens so that much larger log-ratios do not improve or decrease the estimation quality. This means that, similar to the FOAW results, there is no valley in the error surface that can be used for automatic optimisation; rather, we must select a reasonably low set of values such that error and delay performance is near-optimal.

We can conclude that as the error surface approaches its minimum in the top-right corner, a *sufficient* log-ratio relative to the position covariance R_p is reached, and we can then independently examine the relationship between R_a and Q in that region.

From B.10(a), we can observe that for low friction gain, there is slightly lower error below the diagonal, i.e. when Q has a higher value than R_a . (Since we have defined our log-ratios as $\log R_p / \log x$, higher log-ratios coincide with higher absolute covariance.) There is also a sub-region in the lower part of the graph, coloured white, where high Q values lead to divergence.

As gain increases, the preference for lower R_a evolves into a striated shape in the lower-right diagonal of the graph, where $Q > R_a$. This is reminiscent of the striped pattern found in the complementary filter surface. Lower error can be found in the thin horizontal regions where Q is able to increase seemingly arbitrarily. The top-most stripe nearest the diagonal is the widest, and seems to be stable across all our higher-gain examples.

References

- [1] J. Abbott and A. Okamura. Effects of position quantization and sampling rate on virtual-wall passivity. *Robotics, IEEE Transactions on*, 21(5):952–964, Oct. 2005.
- [2] R. Adams and B. Hannaford. Control law design for haptic interfaces to virtual reality. *IEEE Transactions on Control Systems Technology*, 10(1):3–13, 2002.
- [3] R. J. Adams and B. Hannaford. Stable haptic interaction with virtual environments. *IEEE Transactions on Robotics and Automation*, 15(3):465–474, 1999.
- [4] J. M. Adrien. *Étude de structures complexes vibrantes, applications à la synthèse par modèle physique*. PhD thesis, Université Paris 6, 1988.
- [5] K. Altman. How to choose a bow, 2006. URL: http://www.altmanbows.com/how_to_choose_a_bow.html. Accessed July 2012.
- [6] A. Askenfelt. Measurement of the bowing parameters in violin playing. II: Bow-bridge distance, dynamic range, and limits of bow force. *J. Acoust. Soc. Am.*, 86(2):503–516, August 1989.
- [7] K. Åström. Control of systems with friction. In *Proceedings of the Fourth International Conference on Motion and Vibration Control*, Zurich, Aug. 1998.
- [8] Bar-Shalom, Yaakov, Kirubarajan, Thiagalingam, Li, and X.-Rong. *Estimation with Applications to Tracking and Navigation*. John Wiley & Sons, Inc., New York, NY, USA, 2002.
- [9] S. Barrass and M. Adcock. Interactive granular synthesis of haptic contact sounds. In *Proceedings of the AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio*, pages 270–277, Helsinki University of Technology, Espoo, Finland, June 2002.
- [10] P. Bélanger, P. Dobrovolny, A. Helmy, and X. Zhang. Estimation of angular velocity and acceleration from shaft-encoder measurements. *The International Journal of Robotics Research*, 17(11):1225–1233, 1998.

-
- [11] P. D. Bennett. Hap-Kit: A haptic interface for a virtual drum-kit. In *The Symposium for Cybernetics Annual Research Projects*, University of Reading, June 2004.
- [12] P. D. Bennett, N. Ward, S. O’Modhrain, and P. Rebelo. DAMPER: a platform for effortful interface development. In *Proceedings of the Conference on New Interfaces for Musical Expression*, pages 273–276, New York, NY, June 2007.
- [13] E. Berdahl, G. Niemeyer, and J. Smith, III. Using haptic devices to interface directly with digital waveguide-based musical instruments. In *Proceedings of the Ninth International Conference on New Interfaces for Musical Expression, Pittsburgh, PA*, pages 183–186, 2009.
- [14] G. Bishop and G. Welch. An introduction to the kalman filter. In *SIGGRAPH 2001 Course Notes*, 2001.
- [15] I. Brouwer, K. MacLean, and A. Hodgson. Simulating cheap hardware: A platform for evaluating cost-performance trade-offs in haptic hardware design. In *Proc. ICRA*, pages 770–775. IEEE, 2004.
- [16] J. Brown and J. Colgate. Minimum mass for haptic display simulations. In *Proceedings of the 1998 ASME International Mechanical Engineering Congress and Exhibition*, page 249, Anaheim, California, November 1998.
- [17] R. H. Brown, S. C. Schneider, and M. G. Mulligan. Analysis of algorithms for velocity estimation from discrete position versus time data. *IEEE Trans. on Indus. Elec.*, 39(1):11–19, Feb. 1992.
- [18] G. C. Burdea. *Force and touch feedback for virtual reality*. John Wiley & Sons, Inc., New York, NY, USA, 1996.
- [19] C. Cadoz. *Les réalités virtuelles: un exposé pour comprendre, un essai pour réfléchir*, volume 32. Flammarion, 1994.
- [20] C. Cadoz and J. Florens. Fondement d’une démarche de recherche informatique/musique. *Revue d’Acoustique*, 45:86–101, 1978.
- [21] C. Cadoz and M. M. Wanderley. Gesture-music. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*, pages 71–94. IRCAM, Paris, France, 2000.
- [22] C. Cadoz, A. Luciani, J. Florens, C. Roads, and F. Chadabe. Responsive input devices and sound synthesis by stimulation of instrumental mechanisms: The Cordis system. *Comp. Music J.*, 8(3):60–73, 1984.

-
- [23] C. Cadoz, A. Luciani, and J. L. Florens. CORDIS-ANIMA: a modeling and simulation system for sound and image synthesis—the general formalism. *Computer Music Journal*, 17(1):19–29, 1993. MIT Press.
- [24] G. Campion and V. Hayward. Fundamental limits in the rendering of virtual haptic textures. In *WHC '05: Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 263–270, Pisa, Italy, 2005. IEEE Computer Society.
- [25] N. Castagné and C. Cadoz. Creating music by means of ‘physical thinking’: the musician oriented genesis environment. In *Proceedings of the Conference on Digital Audio Effects*, Hamburg, Germany, 2002.
- [26] C. Chafe. Tactile audio feedback. In *Proceedings of the International Computer Music Conference*, pages 76–79. ICMA, 1993.
- [27] S. P. Chan. Velocity estimation for robot manipulators using neural network. *Journal of intelligent & robotic systems*, 23(2):147–163, 1998.
- [28] V. Chawda, O. Celik, and M. K. O’Malley. Application of Levant’s differentiator for velocity estimation and increased Z-width in haptic interfaces. In *Proc. World Haptics*, Istanbul, Turkey, June 2011.
- [29] L. L. Chu. Haptic feedback in computer music performance. In *Proceedings of the International Computer Music Conference*, pages 57–58. ICMA, 1996.
- [30] F. J. Clark. Neural mechanisms of kinesthesia. In *Human Machine Interfaces for Teleoperators and Virtual Environments*, Santa Barbara, California, Mar. 1990.
- [31] J. E. Colgate and J. M. Brown. Factors affecting the z-width of a haptic display. In *IEEE Conference on Robotics and Automation*, pages 3205–3210, 1994.
- [32] J. E. Colgate and G. Schenkel. Passivity of a class of sample data systems: Application to haptic interfaces. In *Proceedings of the American Control Conference*, pages 3236–3240, 1994.
- [33] J. E. Colgate, M. C. Stanley, and J. M. Brown. Issues in the haptic display of tool use. In *IROS '95: Proceedings of the International Conference on Intelligent Robots and Systems-Volume 3*, pages 3140–3145, Washington, DC, USA, 1995. IEEE Computer Society.
- [34] P. R. Cook. *Real Sound Synthesis for Interactive Applications*. A. K. Peters, Ltd., Natick, MA, USA, 2002.

-
- [35] P. R. Cook and G. Scavone. The synthesis toolkit (STK). In *Proceedings of the International Computer Music Conference*, pages 164–166, 1999.
- [36] P. R. Dahl. Solid friction damping of mechanical vibrations. *AIAA Journal*, 14(12): 1675–1682, 1976.
- [37] M. Demoucron, A. Askenfelt, and R. Causse. Measuring bow force in bowed string performance: Theory and implementation of a bow force sensor. *Acta Acustica united with Acustica*, 95(4):718–732, 2009.
- [38] N. Diolaiti, G. Niemeyer, F. Barbagli, J. K. S. Jr., and C. Melchiorri. The effect of quantization and coulomb friction on the stability of haptic rendering. In *Proc. World Haptics Conf.*, pages 237–246, 2005.
- [39] G. Eckel, F. Iovino, and R. Caussé. Sound synthesis by physical modelling with modalys. In *Proceedings of the International Symposium on Musical Acoustics*, Le Normant, Dourdan, France, 1995.
- [40] J. Florens and C. Cadoz. Instrumental gesture, computer and retroactive gestic transducers. In *International Computer Music Conference*, 1984.
- [41] J. Florens, A. Razafindrakoto, A. Luciani, C. Cadoz, et al. Optimized real-time simulation of objects for musical synthesis and animated image synthesis. In *International computer music conference 1986 proceedings*, pages 65–70, 1986.
- [42] J. Florens, A. Luciani, C. Cadoz, and N. Castagné. ERGOS: Multi-degrees of freedom and versatile force-feedback panoply. In *Proceedings of Eurohaptics*, pages 356–360, 2004.
- [43] J.-L. Florens. Expressive bowing on a virtual string instrument. In A. Camurri and G. Volpe, editors, *Lecture Notes in Computer Science: Gesture-Based Communication in Human-Computer Interaction, 5th International Gesture Workshop*, volume 2915, pages 487–496. Springer Berlin / Heidelberg, Genova, Italy, April 2003.
- [44] A. Freed, X. Rodet, and P. Depalle. Synthesis and control of hundreds of sinusoidal partials on a desktop computer without custom hardware. In *Int. Conf. Signal Processing Applications and Technology*, volume 2, pages 1024–30, Santa Clara, CA, Apr. 1993. DSP Associates.
- [45] B. Gillespie. Dynamical modeling of the grand piano action. In *Proceedings of the International Computer Music Conference. ICMA*, 1992.
- [46] B. Gillespie. *Haptic Display of Systems with Changing Kinematic Constraints: The Virtual Piano Action*. PhD thesis, Stanford University, January 1996.

-
- [47] R. B. Gillespie and M. Cutkosky. Interactive dynamics with haptic display. In *Proceedings of the ASME Winter Annual Meeting*, volume 49, pages 65–72, New Orleans, LA, November 1993. Advances in Robotics, Mechatronics and Haptic Interfaces.
- [48] R. B. Gillespie and M. R. Cutkosky. Stable user-specific haptic rendering of the virtual wall. In *Proceedings of the International Mechanical Engineering Congress and Exhibition*, pages 397–406, Atlanta, GA, 1996. ASME.
- [49] M. Giordano, S. Sinclair, and M. M. Wanderley. Bowing a vibration-enhanced force feedback device. In *Proc. Conference on New Interfaces for Musical Expression*, Ann Arbor, MI, 2012.
- [50] A. H. C. Gosline and V. Hayward. Time-domain passivity control of haptic interfaces with tunable damping hardware. In *WHC '07: Proceedings of the Second Joint EuroHaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 164–179, Washington, DC, USA, 2007. IEEE Computer Society.
- [51] A. H. C. Gosline and V. Hayward. Dual-channel haptic synthesis of viscoelastic tissue properties using programmable eddy current brakes. *The International Journal of Robotics Research*, May 2009.
- [52] B. Hannaford and J. Ryu. Time-domain passivity control of haptic interfaces. *IEEE Transactions on Robotics and Automation*, 18(1):1–10, 2002.
- [53] W. S. Harwin and N. Melder. Improved haptic rendering for multi-finger manipulation using friction cone based god-objects. In *Proc. Eurohaptics*, Edinburgh, U.K., 2002.
- [54] V. Hayward. Haptic synthesis. In *Proc. SYROCO 2006, 8th International IFAC Symposium on Robot Control*, 2006. (Keynote Paper).
- [55] V. Hayward and B. Armstrong. A new computational model of friction applied to haptic rendering. In *Proceedings of The Sixth International Symposium on Experimental Robotics VI*, pages 403–412, London, UK, March 2000. Springer Verlag.
- [56] V. Hayward and O. R. Astley. Performance measures for haptic interfaces. In G. Giralta and G. Hirzinger, editors, *Robotics Research: The 7th International Symposium*, pages 195–207, Berlin, Germany, 1996. Springer Verlag.
- [57] V. Hayward and K. MacLean. Do it yourself haptics: part I. *Robotics & Automation Magazine*, 14(4):88–104, 2007.
- [58] H. L. F. Helmholtz. *On the Sensations of Tone, 4th ed.* Dover, NY, 1954. Die Lehre von den Tonempfindungen, Trans. Ellis, A. J. 1877.

- [59] D. Howard and S. Rimell. Cymatic: A tactile controlled physical modelling instrument. In *Proceedings of the 6th International Conference on Digital Audio Effects*, volume 11, London, UK, Sept. 2003.
- [60] Z. Jabbour, S. Moreau, A. Riwan, F. Khatounian, and G. Champenois. Speed estimation improvement using full order state observer for a haptic interface. In *IEEE Int. Conf. on Industrial Technology*, pages 1–5. IEEE, 2009.
- [61] F. Janabi-Sharifi, V. Hayward, and C.-S. J. Chen. Discrete-time adaptive windowing for velocity estimation. *IEEE Trans. on Cont. Sys. Tech.*, 8(6):1003–1009, Nov. 2000.
- [62] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.
- [63] D. Karnopp. Computer simulation of stick-slip friction in mechanical dynamic systems. *ASME Journal of Dynamic Systems, Measurement and Control*, 1985.
- [64] D. Kaufman. Science project-the Henson company uses its Henson digital puppetry studio to create real-time animation within a live-action type of setting for the new children’s television series Sid the Science Kid. *Computer Graphics World*, 31(9):28, 2010.
- [65] E. Kilic, O. Baser, M. Dolen, and E. I. Konukseven. An enhanced adaptive windowing technique for velocity and acceleration estimation using incremental position encoders. In *Proc. Int. Conf. on Sig. and Elec. Sys.*, pages 61–64, Gliwice, Poland, Sept. 2010.
- [66] H. Kim, J. Choi, and S. Sul. Accurate position control for ac servo motor using novel speed estimator. In *Proc. Int. Conf. Industrial Electronics, Control, and Instrumentation*, volume 1, pages 627–632. IEEE, 1995.
- [67] I. Kim and O. De Weck. Adaptive weighted-sum method for bi-objective optimization: Pareto front generation. *Structural and Multidisciplinary Optimization*, 29(2):149–158, 2005.
- [68] S. Kim, X. Zhang, and Y. Kim. Haptic puppetry for interactive games. *Technologies for E-Learning and Digital Entertainment*, pages 1292–1302, 2006.
- [69] K. Kuchenbecker. *Characterizing and controlling the high-frequency dynamics of haptic interfaces*. PhD thesis, Stanford University, 2006.
- [70] K. Kuchenbecker, J. Fiene, and G. Niemeyer. Event-based haptics and acceleration matching: Portraying and assessing the realism of contact. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 381–387, 2005.

-
- [71] D. Lawrence and J. Chapel. Performance trade-offs for hand controller design. In *Proc. International Conference on Robotics and Automation*, pages 3211–3216. IEEE, 1994.
- [72] A. Levant. Robust exact differentiation via sliding mode technique. *Automatica*, 34(3):379–384, 1998.
- [73] A. Levant. Principles of 2-sliding mode design. *Automatica*, 43(4):576–586, 2007.
- [74] Y.-A. Lim, H. S. Ahn, and J. Ryu. Performance of equivalent frequency-dependent damping. In *WHC '09: Proceedings of the World Haptics 2009 - Third Joint EuroHaptics conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 559–564, Washington, DC, USA, 2009.
- [75] S. H. Linkwitz. Active crossover networks for noncoincident drivers. *J. Audio Eng. Soc.*, 24(1):2–8, 1976.
- [76] A. Luciani, J.-L. Florens, and N. Castagné. From action to sound: a challenging perspective for haptics. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC'05)*, pages 592–595, Pisa, Italy, March 2005. IEEE Computer Society.
- [77] R. Marler and J. Arora. Survey of multi-objective optimization methods for engineering. *Structural and multidisciplinary optimization*, 26(6):369–395, 2004.
- [78] T. H. Massie and J. K. Salisbury. The PHANTOM haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Chicago, IL, Nov 1994.
- [79] M. McIntyre, R. Schumacher, and J. Woodhouse. Aperiodicity in bowed-string motion. *Acustica*, 49(13), 1981.
- [80] M. E. McIntyre, R. Schumacher, and J. Woodhouse. On the oscillations of musical instruments. *The Journal of the Acoustical Society of America*, 74:1325, 1983.
- [81] W. McMahan and K. Kuchenbecker. Haptic display of realistic tool contact via dynamically compensated control of a dedicated actuator. In *Int. Conf. on Intelligent Robots and Systems*, pages 3170–3177. IEEE, 2009.
- [82] M. Minsky, O.-Y. Ming, O. Steele, F. P. Brooks Jr., and M. B. v. Feeling and seeing: issues in force display. In *SI3D '90: Proceedings of the Symposium on Interactive 3D Graphics*, pages 235–241, New York, NY, USA, 1990. ACM Press.

-
- [83] W. Moss, H. Yeh, J. Hong, M. Lin, and D. Manocha. Sounding liquids: Automatic sound synthesis from fluid simulation. *ACM Transactions on Graphics (TOG)*, 29(3):21, 2010.
- [84] I. J. Nagrath, S. N. Sharan, and R. Ranjan. *Signals and Systems*. Tata McGraw Hill, New Delhi, 2nd edition, 2009.
- [85] C. Nichols. The vBow: development of a virtual violin bow haptic human-computer interface. In *Proceedings of the Conference on New Interfaces for Musical Expression*, pages 29–32, Dublin, Ireland, 2002.
- [86] D. Nyce. *Linear position sensors: theory and application*. Wiley-IEEE, 2004.
- [87] I. Oakley, M. R. McGee, S. Brewster, and P. Gray. Putting the feel in ‘look and feel’. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 415–422, The Hague, The Netherlands, 2000.
- [88] T. Ohmae, T. Matsuda, K. Kamiyama, and M. Tachikawa. A microprocessor-controlled high-accuracy wide-range speed regulator for motor drives. *IEEE Transactions on Industrial Electronics*, IE-29(3):207–211, 1982.
- [89] S. M. O’Modhrain. *Playing by Feel: Incorporating Haptic Feedback into Computer-Based musical Instruments*. PhD thesis, Stanford University, November 2000.
- [90] S. M. O’Modhrain and C. Chafe. Incorporating haptic feedback into interfaces for music applications. In *Proceedings of the International Symposium on Robotics with Applications, World Automation Conference*, 2000.
- [91] R. L. Ott and M. Longnecker. *An introduction to statistical methods and data analysis*. Brooks/Cole Cengage Learning, Belmont, CA, USA, 6th edition, 2010.
- [92] R. Pallas-Areny and J. G. Webster. *Sensors and signal conditioning; 2nd ed.* Wiley, New York, NY, 2001.
- [93] V. Pareto. *Manuale di Economica Politica, Societa Editrice Libraria*. Milan, 1906. Translated into English by A.S. Schwier as *Manual of Political Economy*, edited by A.S. Schwier and A.N. Page, 1971. New York: A.M. Kelley.
- [94] E. A. Parr. *Industrial Control Handbook; 3rd ed.* Industrial Press, New York, NY, 2000.
- [95] N. Patel, R. Smith, and Z. Zabinsky. Pure adaptive search in monte carlo optimization. *Mathematical programming*, 43(1):317–328, 1989.

-
- [96] R. Pitteroff. Modelling of the bowed string taking into account the width of the bow. In *Stockholm Musical Acoustics Conference (SMAC-93)*, pages 407–410, Stockholm, July 1993. Royal Swedish Academy of Music.
- [97] R. Pitteroff and J. Woodhouse. Mechanics of the contact area between a violin bow and a string. part I: Reflection and transmission behaviour. *Acta Acustica united with Acustica*, 84(4):543–562, 1998.
- [98] R. Pitteroff and J. Woodhouse. Mechanics of the contact area between a violin bow and a string. part II: Simulating the bowed string. *Acta Acustica united with Acustica*, 84(4):744–757, 1998.
- [99] R. Pitteroff and J. Woodhouse. Mechanics of the contact area between a violin bow and a string. part III: Parameter dependence. *Acta Acustica united with Acustica*, 84(5):929–946, 1998.
- [100] M. Puckette. *The Theory and Technique of Electronic Music*. World Scientific Press, Singapore, 2007.
- [101] K. J. Åström and B. Wittenmark. *Computer Controlled Systems: Theory and Design*. Prentice Hall Professional Technical Reference, 1984.
- [102] K. J. Åström, P. Albertos, M. Blanke, and A. Isidori. *Control of Complex Systems*. Springer, 2000.
- [103] C. Richard. *On the identification and haptic display of friction*. PhD thesis, Stanford University, 2000.
- [104] C. Richard and M. R. Cutkosky. Friction modeling and display in haptic applications involving user performance. In *Proceedings of the IEEE International Conference on Robotics and Automation*, volume 1, pages 605–611, 2002.
- [105] C. Richard, M. Cutkosky, and K. MacLean. Friction identification for haptic display. In *Proceedings of the ASME Dynamic Systems and Control Division*, volume 67, pages 327–334, 1999.
- [106] G. Robles-De-La-Torre. The importance of the sense of touch in virtual and real environments. *IEEE Multimedia*, 13(3):24–30, 2006. Special issue on Haptic User Interfaces for Multimedia Systems.
- [107] M. Sakaguchi, J. Furusho, and N. Takesue. Passive force display using ER brakes and its control experiments. In *Proc. Virtual Reality*, pages 7–12. IEEE, 2001.
- [108] E. Samur. *Performance Metrics for Haptic Interfaces*. Springer Series on Touch and Haptic Systems. Springer London, 2012.

-
- [109] F. Schoen. Stochastic techniques for global optimization: A survey of recent advances. *Journal of Global Optimization*, 1(3):207–228, 1991.
- [110] E. Schoonderwaldt, S. Sinclair, and M. M. Wanderley. Why do we need 5-DOF force feedback? The case of violin bowing. In *Proceedings of Enactive '07*, pages 397–400, 2007.
- [111] R. Schumacher, S. Garoff, and J. Woodhouse. Probing the physics of slip-stick friction using a bowed string. *The Journal of Adhesion*, 81(7):723–750, 2005.
- [112] E. Scilingo, A. Bicchi, D. De Rossi, and A. Scotto. A magnetorheological fluid as a haptic display to replicate perceived biological tissues compliance. In *Int. Conf. Microtechnologies in Medicine and Biology*, pages 229–233. IEEE, 2000.
- [113] D. Senkal, H. Gurocak, and E. Konukseven. Passive haptic interface with mr-brakes for dental implant surgery. *Presence: Teleoperators and Virtual Environments*, 20(3):207–222, 2011.
- [114] S. Serafin and J. O. Smith, III. A multirate, finite-width, bow-string interaction model. In *Proceedings of the COST-G6 Conference on Digital Audio Effects (DAFx)*, pages 207–210, Verona, Italy, 2000.
- [115] S. Serafin, J. O. Smith III, and J. Woodhouse. An investigation of the impact of torsion waves and friction characteristics on the playability of virtual bowed strings. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 87–90, 1999.
- [116] S. Serafin, F. Avanzini, and D. Rocchesso. Bowed string simulation using an elasto-plastic friction model. In *Proc. Stockholm Music Acoustics Conf.*, pages 95–98, 2003.
- [117] S. Sinclair and M. M. Wanderley. A run-time programmable simulator to enable multi-modal interaction with rigid-body systems. *Interact. Comput.*, 21(1-2):54–63, 2009.
- [118] S. Sinclair, G. Scavone, and M. M. Wanderley. Audio-haptic interaction with the digital waveguide bowed string. In *Proc. of the International Computer Music Conference*, pages 275–278, Montreal, Canada, 2009.
- [119] S. Sinclair, J.-L. Florens, and M. M. Wanderley. A haptic simulator for gestural interaction with the bowed string. In *10^{ème} Congrès Français d’Acoustique*, Lyon, France, Apr. 2010. SFA.
- [120] S. Sinclair, M. M. Wanderley, V. Hayward, and G. Scavone. Noise-free haptic interaction with a bowed-string acoustic model. In *Proc. World Haptics Conf.*, Istanbul, Turkey, July 2011.

-
- [121] J. Smith. Principles of digital waveguide models of musical instruments. *Applications of digital signal processing to audio and acoustics*, pages 417–466, 2002.
- [122] J. Smith and J. Woodhouse. The tribology of rosin. *Journal of the Mechanics and Physics of Solids*, 48(8):1633–1682, 2000.
- [123] J. O. Smith. Efficient simulation of the reed-bore and bow-string mechanisms. In *Proc. of the Int. Comp. Music Conf.*, pages 275–280, The Hague, 1986. CMA.
- [124] J. O. Smith. Physical modeling using digital waveguides. *Comp. Mus. J.*, 16(4): 74–91, 1992.
- [125] J. O. Smith. *Physical audio signal processing: For virtual musical instruments and audio effects*. W3K Publishing, 2010. Available online: <http://ccrma.stanford.edu/~jos/pasp/>.
- [126] H. Sussmann and P. Kokotovic. The peaking phenomenon and the global stabilization of nonlinear systems. *IEEE Transactions on Automatic Control*, 36(4):424–440, 1991.
- [127] H. Z. Tan, M. A. Srinivasan, B. Eberman, and B. Cheng. Human factors for the design of force-reflecting haptic interfaces. *Dynamic Systems and Control*, 55(1), 1994.
- [128] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen. Discrete-time modelling of musical instruments. *Reports on progress in physics*, 69:1, 2006.
- [129] K. van den Doel, P. G. Kry, and D. K. Pai. FoleyAutomatic: physically-based sound effects for interactive simulation and animation. In *Proceedings of the 28th annual conference on computer graphics and interactive techniques*, pages 537–544, 2001.
- [130] H. Vasudevan, M. B. Srikanth, and M. Muniyandi. Rendering stiffer walls: a hybrid haptic system using continuous and discrete time feedback. *Advanced Robotics*, 21: 1323–1338, Nov. 2007.
- [131] R. T. Verrillo. Vibration sensing in humans. *Music Perception*, 9(3):281–302, 1992.
- [132] D. Weir, J. Colgate, and M. Peshkin. Measuring and increasing z-width with active electrical damping. In *Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 169–175, Reno, NE, Mar. 2008.
- [133] J. Woodhouse. Physical modeling of bowed strings. *Computer Music Journal*, 16(4): 43–56, 1992.
- [134] J. Woodhouse. Bowed string simulation using a thermal friction model. *Acta Acustica united with Acustica*, 89:355–368, 2003.

-
- [135] J. Woodhouse and P. Galluzzo. The bowed string as we know it today. *Acta Acustica united with Acustica*, 90:579–589, July 2004.
- [136] J. Woodhouse, R. Schumacher, and S. Garoff. Reconstruction of bowing point friction force in a bowed string. *J. Acoustical Society of America*, 108:357–368, 2000.
- [137] Z. Zabinsky. Stochastic search methods for global optimization. *Wiley Encyclopedia of Operations Research and Management Science*, 2010.
- [138] C. Zilles and J. Salisbury. A constraint-based god-object method for haptic display. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 3, pages 3146–3151, 1995.