A Consort of Gestural Musical Controllers: Design, Construction, and Performance

Joseph W. Malloch

Music Technology Area
Schulich School of Music
McGill University
Montreal, Canada

February 2008

A thesis submitted to McGill University in partial fulfillment of the requirements for the degree of Master of Arts.

2008/02/02

Abstract

This thesis project presents the T-Sticks, a new family of digital musical instruments (DMIs). Most DMIs are either entirely unique interfaces, or exist as design iterations in which each incarnation is intended to improve on the last. The T-Sticks are instead intended to form a complementary group or consort which may be performed *ensemble* and also performed individually in solo pieces or works for mixed instrumentation. Each of the T-Sticks is based on the same general structure and sensing platform, but each also differs from its siblings in size, weight, timbre and register.

This document explores some of the issues challenging and motivating the field of DMI design and performance, and describes the motivations behind the T-Stick project in this context. Several existing DMIs are examined for similarities to the T-Stick and compared in terms of design intention, implementation, and usage. The hardware and software designed and built for this project is presented, along with insights gained through collaboration with performers and composers in the context of McGill University's *Digital Orchestra* project. The performers in question have collectively practiced and performed with the T-Stick for hundreds of hours in the lab, practice room, and on the concert stage. The consort of T-Sticks will be featured as an ensemble in a piece to be performed during the 2008 MusiMarch festival in Montréal.

Resumé

Ce projet de thèse présente les T-Sticks, une nouvelle famille d'Instruments de Musique Numériques (IMNs). La plupart des IMNs sont soit des interfaces uniques ou des itérations de conception qui se veulent chaque fois une amélioration sur la version précédente. Les T-Sticks forment plutôt un groupe complémentaire qui peut être joué ensemble ainsi qu'individuellement, dans des pièces solo ou des oeuvres pour instrumentation mixte. Chaque T-Stick est basé sur la même structure générale et plate-forme de capteurs, mais chacun diffère de ses frères en dimension, poids, timbre, et registre.

Ce document explore certains défis et motivations du domaine de conception d'IMNs et de performance, et décrit les motivations derrière le projet T-Stick dans ce contexte. Plusieurs IMNs sont examinés et comparés avec les T-Sticks en matière de conception, mise en oeuvre et utilisation. Le matériel et le logiciel conus et construits pour ce projet sont présentés, ainsi que les observations faites lors de collaborations avec des musiciens et compositeurs dans le contexte du projet d'Orchestre Numérique de l'Université McGill. Les musiciens en question ont collectivement pratiqué et joué avec les T-Sticks durant des centaines d'heures dans le laboratoire, la salle de pratique, et sur la scène de concert. Le groupe de T-Sticks sera présenté comme un ensemble dans une pièce qui sera jouée dans le cadre du festival MusiMars en 2008 à Montréal.

Acknowledgments

First of all, many thanks to my advisor Marcelo Wanderley, for guidance, insight, and inspiration. Thanks also to Sean Ferguson, who – with Marcelo – suggested the project that begat the T-Stick, and to D. Andrew Stewart, who began this project with me and contributed unreasonable amounts of time to mapping and synthesis. My colleagues in the McGill Music Technology area have also been a tremendous resource, who never cease to surprise me with their originality and diversity of opinion. Special thanks to Dave Birnbaum, Elliot Sinyor, Steve Sinclair, Mark Marshall, and Darryl Cameron for technical help, discussion and fascinating arguments.

Many thanks to the members of the Digital Orchestra project – D. Andrew Stewart, Fernando Rocha, Xenia Pestova, Stephen Sinclair, Mark Marshall, Heather Hindman, Chloé Dominguez, and Rodolphe Koehly – without whom this project would never have grown to its present state. Thanks are also due to the Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT), for supporting this project financially through student awards.

Finally, a special thanks to my family – Susan, Louise, Nora, and Helen – not only for agreeing to proof-read this thesis, but also for their support and interest over the years.

Contents

List of Figures

List of Tables

List of Acronyms

DMI	Digital Musical Instrument
GUI	Graphical User Interface
HCI	Human-Computer Interaction
HID	Human Interface Device
IC	Integrated Circuit
IDMIL	Input Devices and Music Interaction Laboratory
LED	Light Emitting Diode
OSC	OpenSound Control
PCB	Printed Circuit Board
USB	Universal Serial Bus

Chapter 1

Introduction

Early musical instruments and systems were entirely reliant on mechanical sound production, and usually constructed such that user interface and sound production mechanism were physically the same object. As technological development progressed, however, so did the ability to split the instrument into separate parts, allowing each to be separately optimized [1]. An organ, for example, includes an interface optimized for human fingers and sound producing machinery optimized for range, volume and timbral flexibility. Unlike earlier instruments, the performer does not have to physically touch every part of the instrument in order to activate it. The introduction of electronic instruments in the early twentieth century strengthened this trend greatly: since mechanical linkages were no longer necessary between user interface and sound synthesizer, the control surface could essentially be any arbitrary shape [2]. The move from analog to digital electronics made this arrangement even easier, and the profusion of sensors developed for various industrial and scientific projects were quickly incorporated into interfaces for controlling sound. These "digital musical instruments" (DMIs) are largely defined by the separation of interface and synthesizer, since every connection between performer gesture and synthesizer control must be "mapped" before the instrument can be played [3].

In the commercial world, digital musical instruments are dominated by the piano keyboard interface, a situation much lamented by experimental musicians and musical instrument designers. Outside the commercial sphere, however, there is an exciting community of experimental performers, media artists, and academics exploring the theory and implementation of new interfaces for musical expression [4]. Unfortunately, many of the new devices have a number of limiting characteristics. Even the more robust are often fragile compared to traditional acoustic instruments, and almost always the designer/creator must be present at all demos, practice sessions, and performances to instruct and trouble-shoot. Of course, sometimes the designer *is* the performer, but it is notable that this does not encourage the construction of DMIs that are mature enough to "leave home." Also, non-keyboard digital musical instruments are usually unique, which makes for interesting demos, but how does one learn to play them? Surely the process of learning to play one new interface would help when learning a second, leading to some useful pedagogical generalizations (such as [5]), but what if the next DMI you picked up belonged to the same family as one you already knew how to play?

1.1 Project Overview

This document presents the T-Stick, a new DMI designed to explore and gain insight into some of these issues. Like many DMIs, the T-Stick is intended to foster "expressive" musical performance, be engaging for new users, and rewarding to practice. The T-Sticks differ from other DMIs, however, in that they are intended to form a *family* of instruments, each of which varies from the others while still conceptually belonging to a group. Some new interfaces seem to belong to a family, yet often the members are essentially design iterations, with each one aiming to improve on the last, rather than coequal members of an ensemble or consort.

This thesis project involved building hardware "input devices" and software tools to process sensor data and synthesize sound. The design phase of a project like this necessarily involves research in a large number of disciplines, including gesture, human behavior and perception, human-computer interaction, design aesthetics, and musical performance. Physical construction of the prototype devices was also intended to avoid issues with fragility identified with many new prototypes. Broadly, design and construction of the T-Stick is aimed at producing a digital musical instrument that performers can take away with them, connect with no supervision, and then practice or perform for many hours a day without it wearing out or breaking.

1.2 Thesis Overview

The remainder of this document is organized as follows. Chapter 2 surveys issues necessary for providing context for the thesis project in the areas of instrument classification and gestural control of music. Musical interfaces similar to the one constructed for this thesis are identified and described. Chapter 3 describes the conception, direction, design and construction of the hardware input devices built for this thesis. Chapter 4 details software design and implementation for interfacing with the aforementioned input devices, mapping gestures to sound parameters, and synthesizing sound. Frameworks used in the software design process are also described in detail. Chapter 5 examines and discusses musical performance issues and insights pertaining to practice and performance with the system developed. Finally, Chapter 6 presents conclusions, and points to future work that will continue after this thesis is complete.

1.3 Contributions

The contributions to this thesis work are the exploration of issues related to digital musical instruments and gestural controllers, the design and construction of a family of novel hardware input devices, a "plug-and-play" collaborative mapping system, an OSC-controlled software modal synthesizer for use with the input devices, and this thesis document, which describes and explores the research undertaken.

Chapter 2

Background

This chapter is divided into three main sections. The first section examines the grouping and classification of musical instruments, both traditional and digital, in order to provide context for a discussion on digital musical instrument families. The second section explores important issues related to gestural control of music and digital musical instrument design from existing literature. The third section describes several musical interfaces superficially similar to the one developed for this thesis.

A note on terminology: The title of this thesis uses the term "gestural musical controller" to describe the objects constructed for this project, but it is in fact not my intention to introduce yet another term into the already muddied waters. For the sake of clarity and consistency with other published research, the term "digital musical instrument" will be used to describe the entire system (including mapping and synthesis) and "gestural controller" or "physical interface" will be used to describe the hardware elements of that system.

2.1 Instrument Classification

Traditional acoustic and electric instruments are commonly grouped conceptually into families, based on an examination of their various qualities. In some cases the desire to organize or categorize instruments is purely academic, however important performance practice, pedagogical, and compositional decisions are based on the similarities and differences between instrument families and individual members. Performers may identify themselves as string players or brass players for example, and may perform more than one member of their instrumental group.

This section examines current approaches to instrument classification, and suggests how a family of DMIs might be advantageous.

2.1.1 Classification by Sound

By far the most popular framework used for classification of musical instruments is the Hornbostel-Sachs system, itself based on the earlier Mahillon system [6]. This system initially placed all instruments into four basic groups based on the method of initial sound production. A fifth group was added later.

1. *Idiophones* are instruments in which the sound is produced by the body of the instrument vibrating, such as bells, xylophones, or claves.
2. *Membranophones* are instruments in which the sound is produced by a vibrating membrane, such as drums.
3. *Chordophones* are instruments in which the sound is produced by a vibrating string, such as violins and guitars.
4. *Aerophones* are instruments in which the sound is produced by vibrating air, such as whistles, horns and trumpets.
5. *Electrophones/Electronophones* are instruments in which the sound is produced by electricity.

The diversity of DMIs built over the years clearly calls for a more specific classification system with which to formalize comparisons, since in this system they are almost all relegated to the fifth group, itself not very well defined and the source of some controversy among organologists. Another approach, based on the spectral and temporal features of their sound [7], is also not well suited to distinguishing DMIs with flexible synthesis engines.

2.1.2 Classification by Interface

Mann proposed a system of classification based on the state of matter (solid, liquid, gas, plasma, or "information") associated with both sound production mechanism and the user interface [1]. This system further divides *Gaiaphones* – instruments composed of solids – by the number of dimensions of the vibrating element. At this point the Mann system is still exploratory, and does not offer more detailed categorization.

Confusion often results when categorizing instruments based on user interface, especially when interface and sound production are closely linked. It seems clear that when an organologist speaks of a double reed instrument they are referring to its method of producing sound, but to an oboist the double reed is also part of their user interface. Are the differences between a pianist and an organist primarily related to the sound or to the interface? Wanderley classifies gestural controllers by their resemblance to existing acoustic instruments, as *augmented musical instruments*, *instrument-like gestural controllers*, *instrument-inspired gestural controllers*, and *alternate controllers* [2]. This categorization relates mostly to user interface, but examples in the *augmented musical instruments* blur the line, since they usually retain all of their acoustic sound-producing ability.

2.1.3 Classification by Interaction

A classification of musical instruments is also possible from the point of view of performer-instrument interaction, which forms a broad base of metrics for distinguishing musical instruments. Idiomatic gesture [8, 9], latency and control intimacy [10], and human behavior [11, 12] are all examples applicable to both traditional and digital instruments. In [13], seven different metrics were combined in a dimension-space approach to analyzing and differentiating augmented instruments, DMIs, interactive installations, and musical toys.

2.1.4 Families of Instruments

DMIs are typically unique in both interface and sound production, a situation which has hindered development of performance practice norms and thus any coherent approach to ensemble performance. Some digital musical instruments seem to belong to a family (the *Squeezevoxen* [14]. for example), yet the members are essentially design iterations, each one aiming to improve on the last, rather than coequal members of an ensemble or consort. A family of digital musical instruments would allow experimentation with composition and performance within existing ensemble idioms, but using digital musical instruments.

In the introduction to this section it was posited that important performance practice, pedagogical, and compositional decisions are based on the similarities and differences between instrument families and individual members. It does not necessarily follow, however, that all classification systems will be equally useful in informing these decisions. Although each of the classification approaches takes a dramatically different point of view, the fact that they have different functions has been obscured by the monolithic nature of traditional instruments. When we speak of bowed string instruments, for example, we mean a group of instruments that share a common interface, sound-production method, interaction behavior, and general timbre. As long as the group shares everything in common, it is not necessary to determine which aspects specifically inform composition, or pedagogy, but in the case of DMIs the constituent parts are completely separable. Although it is the aim in this thesis to create a family of DMIs which do share all of these characteristics, it is helpful to consider them separately in order to articulate the advantages such an approach might bring.

Timbre: A family of DMIs as classified by their timbre would be useful for composition, allowing a composer to work within existing consort or ensemble idioms but with digital instruments. The quartet, piano trio, or sort of DMI choir could all be viable ensembles.

Interface: A family of DMIs as classified by their interface could lead to interesting experiments with pedagogy, determining how easily skills developed on one of the family members could be translated to another (perhaps of a different size).

Interaction: One benefit of performing with a consort composed of this type of family is the additional context given to the audience. When first watching a performance with new controllers it can be difficult for some audience members to tell what is going on, or even if the performer is playing skillfully. Allowing them to see several performers playing with roughly the same technique could help shorten or eliminate this period of unfamiliarity.

2.2 Issues in Gestural Control of Music

Since the field's inception, many issues have been raised concerning the design and use of DMIs. These issues have greatly informed the efforts of subsequent instrument designers. Some of these were simply adopted or translated from research in human-computer interaction, while others specifically focus on musical applications. This section will explore some important insights and approaches from past research in gestural control of music.

2.2.1 Interaction Metaphors

One issue faced by both designer and performer of gestural controllers is that connections between gesture and sound must be *designed*, since in the DMI idiom these relationships do not emerge from inherent properties of the building materials and playing technique [3]. While this allows unprecedented flexibility in terms of musical instrument design, the task of designing an instrument that will please performers and audiences can be somewhat daunting. Wessel and Wright propose using explicitly chosen metaphors as a method for organizing and guiding interaction with computer-based instruments [10]. Metaphors discussed include different spatial pitch-arrangements, "timbre space," "drag and drop," scrubbing, "dipping," and "catch and throw." The authors make the important point that metaphors like these determine how we perceive and interact with the world around us (even abstract concepts such as time) and naturally should be considered carefully when designing new systems for control.

In some ways this process can be seen as parallel to the embedding of physical behavior into audio and visual systems in virtual reality or games. There are not actually physical laws at work inside the simulation, but modeling behavior familiar to the user from their experience in the "real world" is logical and provides a basic level of environmental knowledge the user can leverage to explore the system.

2.2.2 Cognitive Load

It is important to consider cognitive load in DMI design, as it is easy for electronic musical instruments to suffer from "feature bloat." Sensors are often cheap, and while it may be desirable to be able to control every parameter of a complex synthesizer in real-time, implementing this control haphazardly can result in chaos. Mulder [15], among others, has advocated employing various levels of data-abstraction for reducing the load on a performer. A careful consideration of the *integrality* or *separability* of both control and synthesis parameters, and mapping them appropriately, can help considerably [16]. In the area of augmented musical instruments, Cook pointed out that some performers may simply lack "spare bandwidth" for additional control, due to the complexity of their current tasks [17].

2.2.3 "Entry Fee" vs. "Ceiling on Virtuosity"

Wessel and Wright consider that the process of "getting started" with a new DMI should be relatively easy, but that this goal should be balanced with allowing users to continually increase their ability through practice [10]. They point out that traditional instruments are often very difficult for a novice to play, and propose that newer interfaces might be designed to allow both low "entry fee" and expansive possibilities for improvement of user skill. The key to achieving this goal is the use of appropriate interaction metaphors for navigation of large or complex parameter-spaces, and very low latency in sensing and signal processing. For reducing latency issues, the authors suggest sampling and communicating sensor data at audio sampling-rates.

2.2.4 Effort

Perhaps the most important DMI design consideration is the requirement for physical effort in performance. In software design it is common to try to reduce the amount of user effort as much as possible, but effort in musical performance is generally seen as a vital ingredient.

> Too often controllers are designed to minimize the physical, chosen <u>because</u> they are effortless. "Effortlessness" is in fact one of the cardinal virtues in the mythology of the computer. [...] Though the principle of effortlessness may guide good word processor design, it has no comparable utility in the design of a musical instrument. In designing a new instrument it might be just as interesting to make playing it as difficult as possible. Physical effort is a characteristic of the playing of all musical instruments. Though traditional instruments have been endlessly refined over the centuries the main motivation has been increase in ranges, accuracy and subtlety of sound and as often as not these are acquired at the expense of comfort. [18]

There is however an important distinction between an expressive performance involving effort and pantomime, as pointed out by Tanaka:

> A virtuoso is often appreciated not for his display of playing technically difficult passages, but more because he is able to play such difficult passages with ease. A theatrical performer who plays an easy passage with an air of difficulty is bound to be less appreciated. [...] Artificially adding physical obstacles

> to require effort would only create the undesired gratuitous effect. [...] It becomes thus a challenge for the instrument builder to conceive of coherent model [sic] for requirement or representation of physical effort on a technology based instrument. [8]

Performance gestures are not all alike: some involve direct, purposeful energy transfer to the instrument or environment which is apparent in the produced sound. Other gestures may damp or modulate the sound dramatically without requiring much effort. In this thesis, the terminology of Cadoz will be used when distinguishing classes of gesture. “Exciter” or “excitation” gestures are those in which the sound energy corresponds to the instrumentalist’s expended energy, or in the case of most acoustic instruments, the instrumentalist is the energy source for sound production. “Modulation” or “modification” gestures modify the instrument’s sonic properties without substantial transfer of energy. “Selection” gestures are used to choose among several similar elements in or on an instrument. [19, 20]

2.3 Similar Interfaces

This section examines several existing interfaces superficially similar to the one described in this thesis. Although all of these interfaces have similar shape, this is not necessarily a meaningful relationship: all are based on cylindrical objects and could just as easily be compared to broomsticks or microphone stands as to each other. A great number of acoustic instruments are also long cylinders, for example flutes, clarinets, bassoons, orchestral chimes, and the didjeridoo.

2.3.1 Sweatstick

Ray Edgar’s *Sweatstick* (figure 2.1) is a long pole-like interface with sliding hand-grips featuring keypads for the fingers [21]. A spring is located in the middle of its length allowing the user to bend the interface, and the distance between hand-grips, their rotation around the rod, and the bend of the interface are all measured [22]. For sensors, Edgar uses ultrasound sensing to determine the distance from the hands to the centre spring section and buttons on the hand grips. The *Sweatstick* was designed and built at STEIM and functions as an alternative MIDI controller for triggering and modifying sounds.

> “The Sweatstick is a one-meter aluminum rod, articulated in the middle, with a sliding keypad for each hand. Performance style variously resembles martial arts, dancing with a broom, hang gliding, praying, and playing guitar.” [23]

Fig. 2.1 The Sweatstick. From http://rayedgar.com/sweatstick.html (Used with permission).

2.3.2 Musical Stick/Talking Stick

Several stick-shaped MIDI controllers were built by Bob Adams at Interval Research, featuring three linear strips for sensing touch position and pressure (figure 2.2). A later version was created for use by Laurie Anderson in her show *Songs and Stories from Moby Dick*. In this incarnation, the interface was renamed the *Talking Stick* and used an array of force-sensing resistors (FSRs) to control granular synthesis [24]. Contributions to later versions of the interface design were also made by IDEO design, and include partial frets between finger positions [25].

2.3.3 The Musicpole

The *Musicpole* (figure 2.3) is a commercial musical interface built by Rags Tuttle for controlling MIDI synthesizers [26]. The device uses a simple sensing method in which the player uses conductive fabric worn on their thumbs to complete a circuit on key-strips wrapped around a short tube. The most innovative aspect of the design is the arrangement of the keys, which are intended to make transposition and certain scale passages much easier to play than on a piano-style keyboard or a string instrument. As such, the *Musicpole* design is intended for triggering MIDI note events, rather than continuous parameter control.

Fig. 2.2 Bob Adams with the first of the sticks he created at Interval Research. Image credit: Bill Verplank (Used with permission).

Fig. 2.3 A photograph of the Musicpole MIDI interface. From http://www.themusicpole.com/ (Used with permission).

Chapter 3

Hardware

Although the physical interface is the most obvious and visible part of a digital musical instrument, the design process begins before the shape, materials, or sensing methods have been decided upon. At the beginning of this project, many of the issues described in the last chapter were considered carefully with the aim that the final instruments, when complete, would remedy some of the deficiencies perceived in other DMIs. To formalize an approach to these issues and organize the design process, a set of design goals was developed.

- **A family of DMIs.** The T-Sticks should form a family analogous to the orchestral string instruments, in which the basic construction, user interface, interaction design, and timbre are the same, but each subclass of T-Stick differs from its siblings in size, weight, and register.

- **A robust physical interface.** The new physical interface should be robust enough that it could be practiced and performed on for hours on end *every day* without breaking or crashing. It is vital that the performers feel that they can work intensively with the interface without fear of breaking it.

- **Simple to operate.** The DMI should be comparable in simplicity to an electric guitar in terms of set-up time and electronic knowledge required. By doing this, the performer will hopefully think of their DMI as they would a traditional instrument, rather than as a "lab experiment."

- **Interaction metaphor.** Interaction with the DMI should be structured as control over a metaphorical vibrating string or bar which can be excited using multiple techniques (striking, bowing, shaking) and damped with nuance. Excitation of sound should require physical energy expenditure by the performer.

- **Integral sensing and mapping.** To achieve this interaction metaphor, sensing and mapping of the performers gestures should be done in an integral, interrelated way such that any manipulation of the physical interface will affect the sound in an intuitive way (as excitation, modification, or selection). Novice interaction should be guided by these relationships, as well as the appearance, feel, and weight of the interface, so that the performer can quickly construct a mental model of the control system.

- **An "expert interface."** The DMI should be designed for the use of expert users, rather than novices. To this end, more emphasis should be put on extending any "ceiling on virtuosity" than on lowering the "entry-fee." New users should be able to produce sound from the DMI, but not necessarily *musical* sound. In the context of [12], the DMI should fit into the *skill/signal* category, exhibit low interruption tolerance, low latency, and maximize the transfer of information (including sensor information) as signals rather than signs, events, or triggers.

- **Performer focus.** The design of the system should aim to keep the performer's focus on the sound and its relation to the entire physical object, rather than individual sensors. Technological concerns should be subsumed under performance or musical concerns (or appear to be such to the performer). As discussed by Rasmussen, the ability to focus on the task rather than the interface may improve with practice [11].

My colleague D. Andrew Stewart had committed to writing pieces for the new DMI. Through discussion of the design goals, a set of physical requirements took shape around which remaining decisions could be made: the physical interface would take the form of an elongated bar or tube, and the design and sensing would be focussed on creating nuanced touch sensing. The interface needed to be able to tell where and how much of it was being touched in multiple areas simultaneously, and it needed to be able to sense the width of a touch, so as to tell the difference between the touch of a finger and that of a hand.

3.1 Differences between T-Sticks

The family of T-Sticks currently exist or are planned in 4 models which vary in size and number of sensors. Some improvements, such as communication speeds or power sources, have changed as the basic design developed, and will eventually be added to the original members to upgrade their performance. Table 3.1 shows the similarities and differences between the members of the T-Stick family.

Table 3.1 Comparison of T-Stick characteristics.

	Register	Length	Diameter	Touch Sensors	Touch Resolution	Frets?	Connection
1	tenor	1.2m	5cm	48	2.54cm	yes	USB
2	alto	0.8m	5cm	32	2.54cm	no	Bluetooth
3	soprano	0.6m	5cm	48	1.27cm	no	Bluetooth
4	bass	1.8m	11.5cm	96	1.27cm	no	serial over USB

3.2 Construction

Fig. 3.1 A view inside the second member of the T-Stick family.

Structurally, the T-Sticks are built using sections of ABS plumbing pipe, with varying diameters and lengths depending on the member of the T-Stick family (see table 3.1 for more information on dimensions). The pipes are cut in half length-wise with a coping saw in order to place sensors, circuitry, and wiring inside. Sensors are affixed to the inside and outside of this base, with wires passing through narrow holes drilled in the plastic where necessary. The whole interface is covered with a protective layer of rubber shrink-tubing to hold it together and add strength (figure 3.2). Solid end-caps cover the ends of the T-Sticks to seal the circuitry and to mount the ports and status LEDs for operating the interface (figure 3.3). The first prototype also features small fret-like protrusions along one of its sensing surfaces, which will be discussed in more detail below. The structural part of the interface is intended to provide a robust platform that protects the sensors from damage and wear; it is strong and the shrink-tubing makes it mostly waterproof. Covering the sensors and circuitry is also intended to make the controller less intimidating for performers who are not technically "savvy," since the only technology apparent is at the most a USB plug and a single power-status LED. If repairs need to be made, the shrink-tubing cover must be sliced and removed; however this is not difficult and should be required less and less often as the T-Stick design gains maturity.

In the initial prototypes, the circuit boards were trimmed to fit tightly inside the diameter of the ABS pipe in order to prevent potential damage to components or movement of the accelerometers. As the circuits have been further developed and made smaller for subsequent versions it has become necessary to pad the interior with open-cell foam thick enough to prevent movement.

Fig. 3.2 The first T-Stick covered, shown before and after shrinking the tubing with a hot air gun.

An adjustable steel spike of 0.5 meters in length was installed in the first T-Stick model,

Fig. 3.3 A view of the endcaps of two T-Stick prototypes, showing the USB ports and status LEDs. Note the hole and fastener for the spike on the right.

enabling the controller to be rested on the floor at a comfortable height for performance in a posture similar to that of a 'cellist. When retracted, the spike slides smoothly into a rubber sheath inside the controller, preventing it from rattling or touching sensors or circuit-boards. Since it adds considerable weight to the controller, the spike is also easily removable if the performer does not wish to play in this way.

3.2.1 Sensors

The T-Stick makes use of a variety of sensors to transduce performer gestures. The remainder of this section explains the choice of sensing methods and details the final implementation.

Multi-touch Position Sensing

The first goal for sensing on the T-Stick was to implement a multi-touch-sensitive outside surface, which at the least had to be able to distinguish between the touch of a finger and that of multiple fingers or a palm. In keeping with the interaction metaphor discussed above, this sensing needed to be unobtrusive, robust, and be able to detect very light touches without deliberately applied pressure.

Touch position sensing is commonly implemented using a *linear position touch sensor* in which the length of a resistive material that must be traveled by electrical current is

Fig. 3.4 The first T-Stick prototype. The integrated spike is visible on the left.

varied according to where the sensor is touched, and the change in resistance is converted to a change in voltage and sampled. [27]. This method only detects a single touch at a time, cannot detect information about the *width* of the touch, and can be very difficult to calibrate finely enough to give accurate position sensing. Instead, it was decided to use an array of simple touch sensors arranged along the length of the physical interface, and to integrate the data afterwards in software. Several options were considered for these touch sensors:

Touch-switch sensing: An extremely simple way to detect touch is to arrange electrical contacts in such a way that a touch completes a circuit, and sample the outputs of the circuits. The connection can be made using a conductive material, as in the *Musicpole* [26], which requires the user to wear "thumbletz" (pieces of conductive fabric worn on the thumbs), or the *Bangarama* [28], which requires the user to wear aluminum foil on the fingers. Another method is to simply use the natural conductance of the player's skin to make the connection, a technique used in Waisvisz' *Crackleboxes* [29], and commonly used in *circuit-bending* [30].

Although simple to implement and generally a reliable way to detect touch, a major disadvantage of this method is that it exposes the electrical contacts (the rudimentary "sensors") to dirt, oxidation, and wear, since they cannot be covered with a protective layer without destroying their sensing ability.

Pressure sensing: Another commonly-used sensor in the DMI community is the force-sensitive resistor (FSR), which changes resistance continuously with pressure applied (up to its maximum). Requiring only a voltage divider per sensor, FSRs are generally considered inexpensive and simple to use. However an array of large numbers of FSRs would in fact be rather expensive and require much more complex wiring than the third option discussed below. Since they are used to sense pressure rather than touch, FSRs could be used to give continuous data about squeezing and "heavy" touches, but experimentation revealed that they were not well-suited for sensing very light touches.

Capacitive touch/proximity sensing: Capacitive sensing is used to measure the capacitance between two or more conductors in a dielectric environment [31]. This type of sensing can be used to measure touch pressure, liquid level or flow, thickness, or position, as well as for vehicle detection [32]. Depending on the implementation, capacitive sensing does not require physical contact for sensing to occur, and can work though various dielectrics including air, plastic, glass, rubber, and wood. Electrodes or antennae used as plates in the capacitive sensor can be any shape or size, though the capacitance produced is proportional to the area of the plates, and inversely proportional to their thickness [27].

For its flexibility and the ability to sense through a protective layer, capacitive sensing was chosen for adding multi-touch capability to the T-Stick.

For the initial prototypes, the Quantum Research Group QProx QT161 charge-transfer capacitive sensing chip was chosen. This type of device is commonly used in consumer electronics to add isometric "button" sensors, especially in devices which are used in potentially dirty environments (e.g. microwaves), since the sensors can be placed behind a seamless protective layer of glass or plastic. The QT161 has a number of advanced features, such as automatic calibration of each channel on power-up, and a "consensus filter" to avoid false detects caused by noise [33]. The QT161 is actually a variation of the QT160 chip, which functions identically to the QT-161 but uses an "adjacent-key suppression" algorithm to avoid detection of touch on adjacent keys simultaneously. For obvious reasons, this function would be incompatible with true multi-touch sensing.

Conductive copper foil tape is used for key electrodes, arranged in strips covering the "front" half of the T-Stick (figure 3.5). For the first two members of the T-Stick family copper tape 12.7mm wide was used, with a 12.7mm gap left between key electrodes, realizing a position-sensing resolution of 25.4mm. The sensing field of each touch sensor is sufficiently large that a touch exactly between copper strips will activate both channels rather than go undetected. On subsequent versions of the T-Stick, the resolution of touch sensing was doubled, using 6.35mm tape and closer spacing between key electrodes.

Fig. 3.5 The first T-Stick prototype before being covered, showing the strips of copper tape used as key electrodes for touch sensing.

Since the field sensitivity of the QT161 ICs must be set using external capacitors, and varies with the size, shape, and thickness of the key electrodes, experiments were performed using a T-Stick mock-up to find the right capacitance. To allow for the thickness of the cover, the field sensitivity was set so that a touch would be detected approximately one millimeter above the surface of the copper tape electrodes. The QT161s are timed using 20MHz crystal oscillators rather than the standard 10MHz in order to increase the responsiveness of the controller.

Vibration/Deformation Sensing

A disadvantage to using discrete touch sensing is the inability to determine the velocity associated with a given touch — information important to sensing the difference between a light touch and a percussive attack. To compensate for the missing attack data, it was decided to use a piezoelectric contact-microphone inside the body of the T-Stick to detect physical vibration and deformation of the base structure. A simple piezoelectric crystal (as commonly used in electric buzzers) is fastened securely to the inside of the front surface of the T-Sticks using epoxy adhesive, and its output connected to signal-processing circuitry. The raw signal is buffered to lower its impedance and clip the voltage to safe levels, since piezoelectric sensors can generate very high voltages. The buffered signal is passed through an envelope-following circuit to convert the audio signal to a corresponding voltage envelope.

Since it is bonded directly to the ABS plastic structure, any deformation of the T-Stick also deforms the piezoelectric disc slightly, causing it to output an electrical signal related to its change in shape. This allows sensing of bending and twisting of the T-Stick, in addition to acoustic vibrations or jarring.

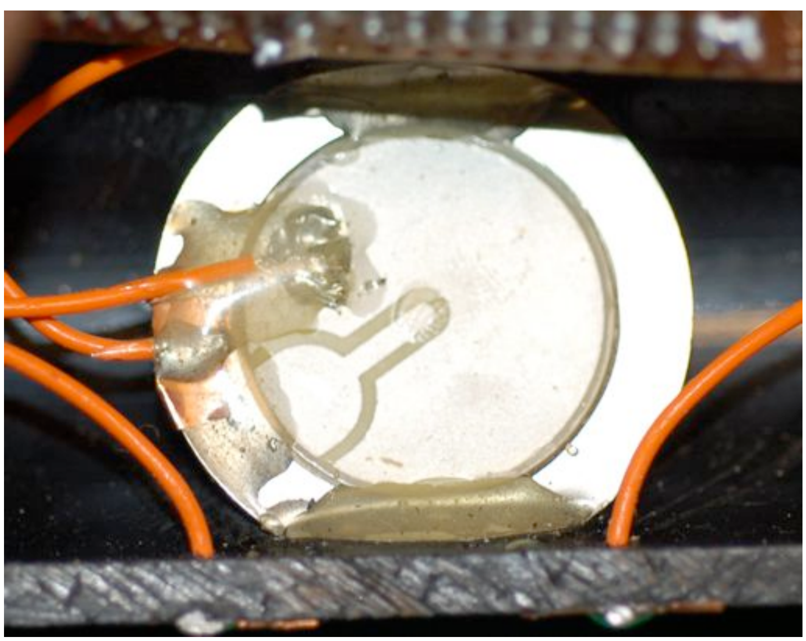

Fig. 3.6 A photograph of a piezoelectric disc bonded to the inside of the T-Stick with epoxy adhesive.

Acceleration Sensing

In order to sense inertial properties of the T-Stick's orientation and movement, the sensor circuit-boards at each end of the controller also include an STMicroelectronics LIS3L02AS4 3-axis 2g/6g linear accelerometer IC, set to 2g sensitivity. One of the axes is left unconnected on one accelerometer, since it is duplicated by the other sensor and is thus redundant. These sensors allow the measurement of the acceleration in three orthogonal axes at each end of the interface, and due to the constant acceleration provided by the Earth's gravity and their sensitivity they also sense inclination or tilt of the interface with respect to the ground. High frequency shaking or vibration of the sensors is also transduced into electrical signal for sampling.

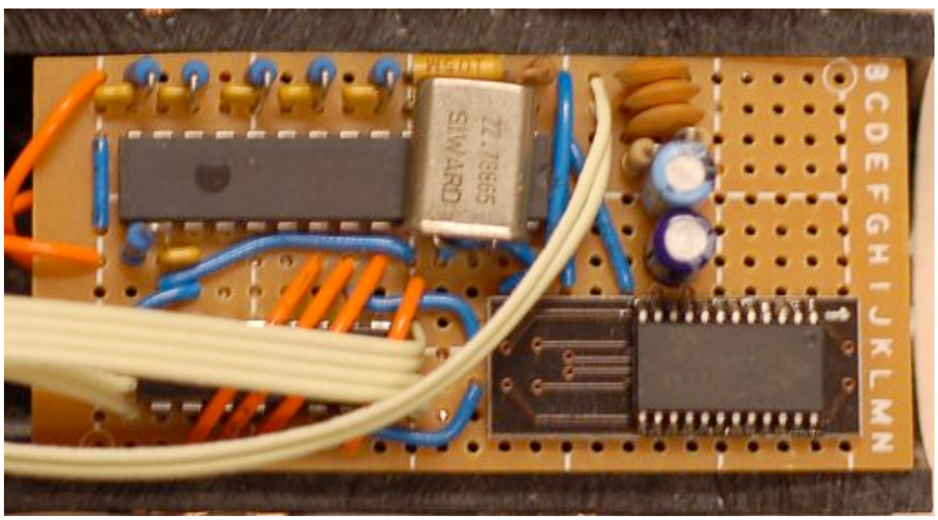

Fig. 3.7 An example of one of the circuit-boards placed at the ends of the early T-Sticks. In addition to capacitive touch-sensing circuitry, this board includes a 3-axis accelerometer, visible on the lower right-hand side.

Pressure Sensing

Two long pressure sensors are fixed to the back of the pipe, with enough space between them and at each end of the controller for a hand to comfortably grasp the pipe without contacting the sensors. The pressure sensors are covered with several layers of thin closed-cell foam to provide proprioceptive feedback ("squishiness") from what are essentially isometric sensors. In the first T-Stick prototype, the pressure sensors used were paper-based sensors custom-built by Rodolphe Koehly [34] which have the advantage of complete flexibility in size and shape and very low cost. Due to frequent repairs, the later T-Sticks use a commercial linear force-sensing resistor (FSR), though there is still interest in using paper sensors as the technology and consistency improves.

One advantage to using shrink-tubing for covering the T-Sticks is that it can be selectively shaped by only applying heat to the areas which should be shrunk. This enables the cover to be very tight and smooth, but still not substantially reduce the range of the pressure sensors.

Fig. 3.8 A view of the back of the first T-Stick, showing the closed-cell foam padding used to cover the pressure sensors.

3.2.2 Data Acquisition

An Atmel ATMEGA16 micro-controller is used for sampling, data formatting, and communication of the sensor data with a laptop computer. The ATMEGA16 includes eight channels of analog-to-digital conversion, which are used for sampling the outputs of accelerometers (five channels), pressure sensors (two channels) and the piezoelectric sensor (one channel). The remaining data from the numerous capacitive sensors outnumbers the digital input pins available on the micro-controller, and two approaches have been taken to acquiring this data. In the first two T-Sticks, digital data from the capacitive sensors is multiplexed using dual four-to-one digital multiplexers, one per capacitive sensing IC, and with two inputs left unconnected. This approach, which effectively implements three-to-one multiplexing, requires only two pins on the micro-controller for control but uses every other available pin for input with a maximum of 48 connected capacitive sensors. This is not coincidentally the number of sensors on the larger of these two T-Stick prototypes.

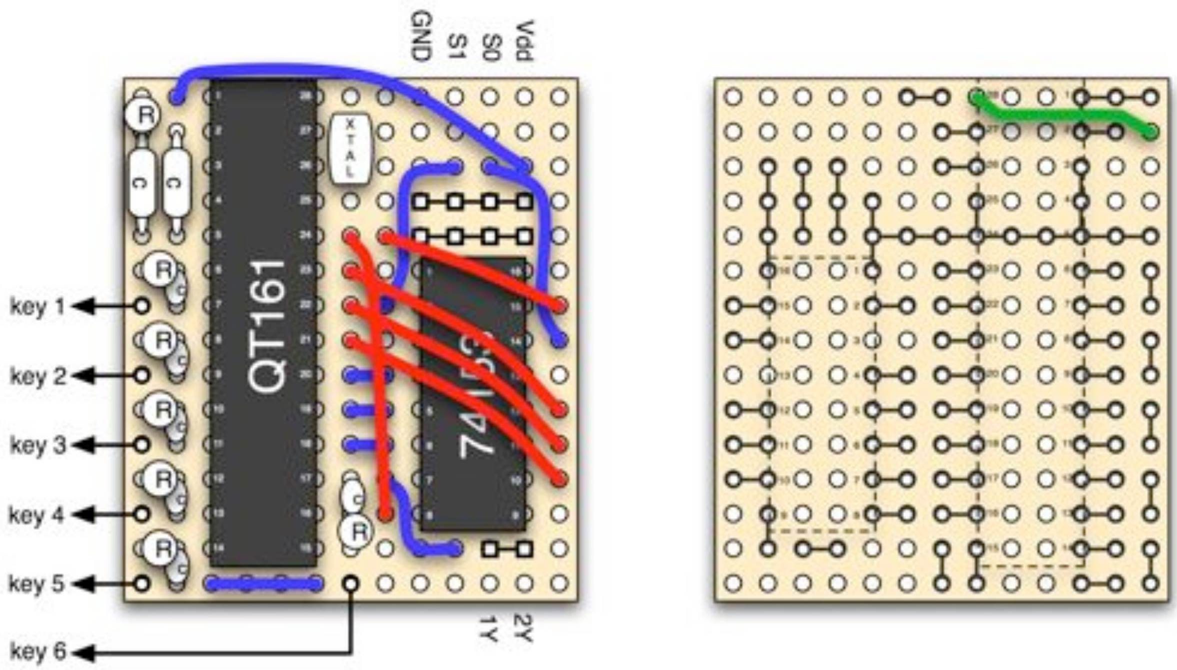

Fig. 3.9 Original circuit-board layout for QT161 capacitive sensing ICs and multiplexers.

In order to increase both the size of the sensing surface and the resolution of sensing, another approach was needed. The solution, shown in figures 3.10 and 3.11 and dubbed “Touch-24” includes four QT-161 ICs along with three 8-stage parallel/serial-input, serial-output shift registers to sense 24 sensor key-electrodes on each board. The printed circuit board (PCB) needs only five connections to the micro-controller, and boards can be “daisy-chained” or connected end to end without any new connections to the micro-controller. The remaining T-Sticks, as well as updates to the original members, will make use of this capacitive-sensing platform, will feature multiples of 24 channels, and will use 6.35mm copper tape for sense electrodes rather than the original 12.7mm tape, effectively doubling the resolution of the sensing surface. By using the new PCB, the amount of wiring is reduced dramatically, making the newer members both tidier and more robust, and much faster to produce. In addition, they are significantly smaller than the old hand-made stripboard versions, allowing them to be used conveniently in higher-resolution sensing implementations (with sense electrodes packed closer together) and also to be placed in smaller-diameter tubes.

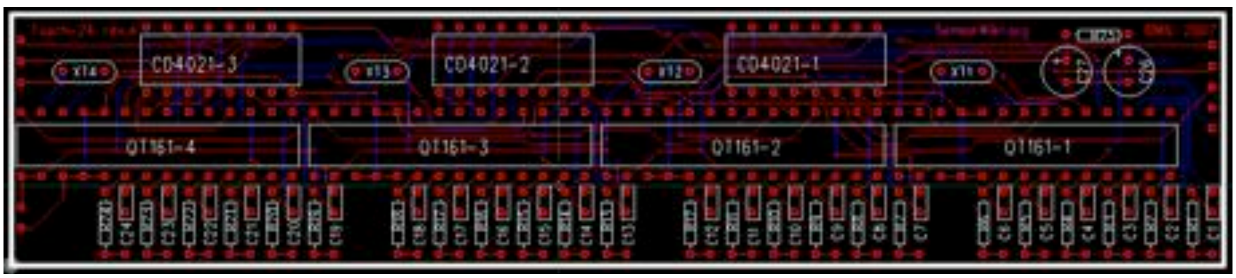

Fig. 3.10 Layout for custom-designed printed circuit-board for sensing 24 channels of capacitive touch data.

3.2.3 Data Processing

Presently, very little data processing is being performed on-board the T-Stick, since it is easier to make quick changes to software during prototyping than it is to re-flash the micro-controller. As the instrument-level mapping matures (see Chapter 4), more data processing and message formatting will be moved onto the interface itself, which will also have the added benefit of reducing the amount of data that needs to be sent to the laptop, allowing the remaining data to be sent at a faster rate. Eventually, sound synthesis itself could be performed on-board, but since the form factor of the T-Sticks does not facilitate the addition of loudspeakers this would not present great advantages.

3.2.4 Communication

The first prototype uses the USB to communicate and to power the onboard electronics [35]; later models use Bluetooth [36] (alto T-Stick) or serial-over USB (bass T-Stick). The newer communication methods allow much faster sensor sampling rates, leading to a more

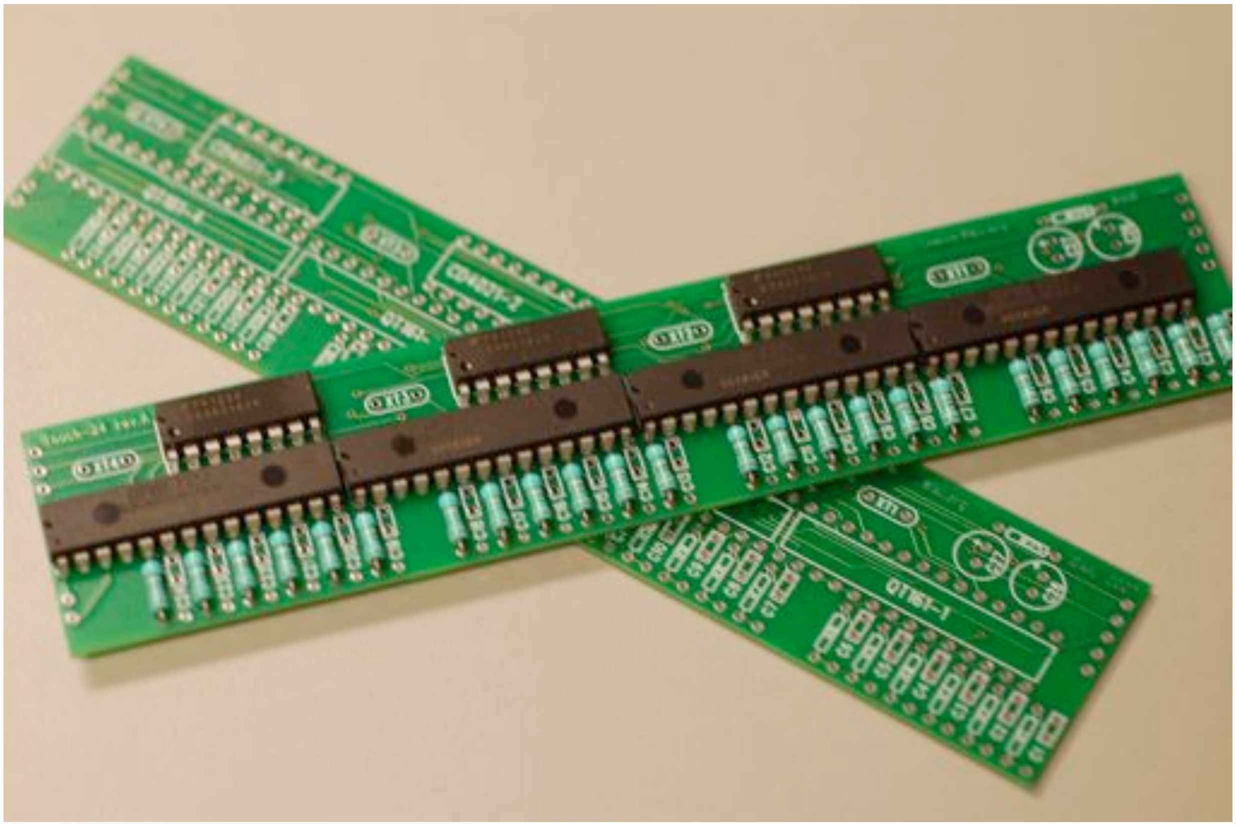

Fig. 3.11 Custom-designed printed circuit-board for sensing 24 channels of capacitive touch data.

responsive-feeling instrument, and, in the case of the Bluetooth models, freedom from constricting wires. The wireless versions obviously require batteries to operate, so a screw-cap is incorporated into one end of the interface for replacing them.

3.3 Summary

In this chapter, the goals of the T-Stick design were outlined, along with a detailed description of the hardware implementation. Basic construction, sensor choice and placement, data acquisition, and communication of the sensor data to a computer were described. Improvements to the electronics platform used were shown, as well as improvements to touch-sensing resolution.

Chapter 4

Software

A framework for software design was developed based on the “abstracted mapping layers” proposed in [37] and [38]. In this framework, mapping connections between an input device and a sound synthesizer are made via abstracted semantic parameters, which allow the user to design high-level relationships between parameters and easily implement a *many-to-many* mapping strategy. It is proposed that all new controllers developed in the Input Devices and Music Interaction Laboratory (IDMIL) use this framework, as it increases the modularity of mapping patches and should make it easier to test new synthesis algorithms with various controllers or to test a new controller with various synthesis algorithms. The software for the T-Sticks was implemented using Max/MSP [39] as modular patches corresponding to the parts of the framework shown in figure 4.1.

4.1 Control Parameters

This part of the software is responsible for connecting to the desired device, receiving sensor data from the hardware, and formatting the data as OSC messages. Depending on the T-Stick member, demultiplexing and/or decoding of the data may also be necessary. HID devices such as the first (tenor) T-Stick are polled for data and respond with each byte preceded by an ID. The Bluetooth and Serial-over-USB devices are configured to require activation, after which they push the sensor data serially, following each cycle with a delimiting character to indicate the end of a line.

The sensor data are smoothed using averaging or median filters, and are scaled to ranges corresponding to physical units wherever possible (e.g. meters) or normalized to fall be-

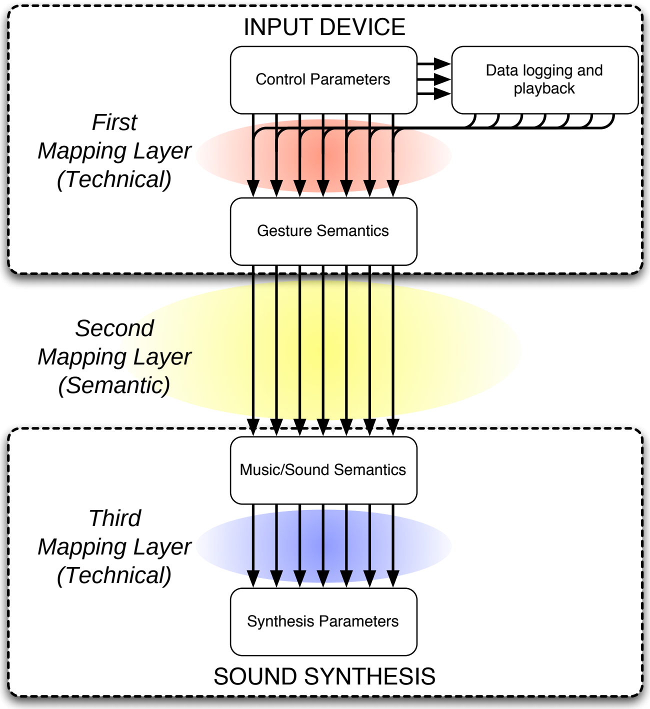

Fig. 4.1 The software framework used for the T-Sticks and other McGill Digital Orchestra DMI's.

tween zero and one. In addition to making parameters easier to understand and map, this approach is designed to allow future improvements to sensor resolution or sampling bit-depth without affecting existing higher level mapping connections. The digital touch-sensing data are translated from decimal form (using 8-bit bytes) into binary format, in which each sensor is represented by a binary digit. All sensor signals are also exposed without alteration for recording or mapping, using the OSC address prefix `/raw`. To distinguish the smoothed and scaled versions of the data, the OSC address prefix `/cooked` is used. The technical control parameters in OSC format are shown in table 4.1.

4.2 First Mapping Layer and Gesture Semantics

The first mapping layer exists within the controller interface, and is concerned with extracting signals or gestures from explicitly designed correlations between the control parameters mentioned above. This corresponds to "body-centred data" as discussed in [40], but since the data are concerned with relationships between the instrument's parts rather than the

Table 4.1 Technical control parameters, with n as the instance ID.

Namespace	Data Type	Units	Min	Max
/tstick/n/raw/piezo	i	n/a	0	255
/tstick/n/raw/pressure/1	i	n/a	0	255
/tstick/n/raw/pressure/2	i	n/a	0	255
/tstick/n/raw/accelerometer/1/x	i	n/a	0	255
/tstick/n/raw/accelerometer/1/y	i	n/a	0	255
/tstick/n/raw/accelerometer/1/z	i	n/a	0	255
/tstick/n/raw/accelerometer/2/x	i	n/a	0	255
/tstick/n/raw/accelerometer/2/y	i	n/a	0	255
/tstick/n/raw/capacitive	i[m/8]	n/a	0	255
/tstick/n/cooked/piezo	f	normalized	0	1
/tstick/n/cooked/pressure/1	f	normalized	0	1
/tstick/n/cooked/pressure/2	f	normalized	0	1
/tstick/n/cooked/accelerometer/1/amplitude	f	normalized	0	1
/tstick/n/cooked/accelerometer/1/angle	f	degrees	0	360
/tstick/n/cooked/accelerometer/1/elevation	f	degrees	-90	90
/tstick/n/cooked/accelerometer/2/amplitude	f	normalized	0	1
/tstick/n/cooked/accelerometer/2/angle	f	degrees	0	360
/tstick/n/cooked/accelerometer/2/elevation	f	degrees	-90	90
/tstick/n/cooked/capacitive	f[m]	normalized	0	1

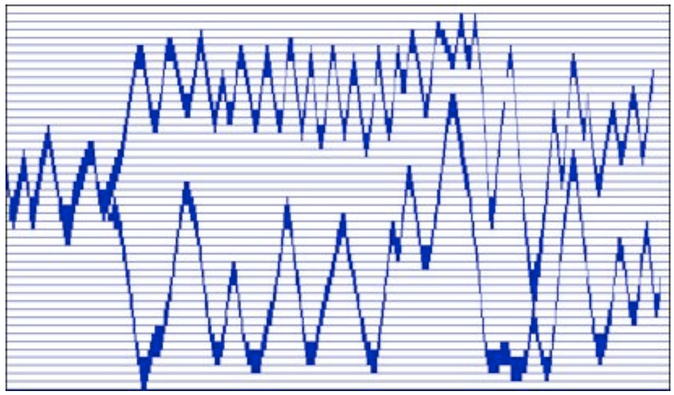

Fig. 4.2 Multislider graph in Max/MSP showing multi-touch data over time: two hands brushing the controller.

player's gesture, and in keeping with the proposed namespace hierarchy in the GDIF project [41], the OSC address prefix `/instrument` is used instead of `/body`.

Instrument relevant data are extracted based on two reference points, corresponding to the two ends of the physical interface. Depending on how the T-Stick is touched or held, a performer can "divide" the interface into smaller sections, and manipulate each section as a subset of the whole. In the following sections the treatment and combinations of control data, and the gestural information extracted from them, are discussed in detail. A diagram is also provided in figure 4.3 to clarify the relationships.

4.2.1 Capacitive Sensors

The capacitive sensor data are provided by the previous software stages as a list of digital values, with each sensor represented by a 1 if the sensor is being touched, or 0 otherwise. These values are integrated over time, with a set decay rate, in order to determine how long each sensor has been touched. Thresholds are used to split the integrated data and to map different parts of the range to different functions. A low value (below the first threshold) is determined to be a quick "brushing" touch and is mapped to excitation based on the area of the interface within the active section being brushed. A value between the first and second thresholds is treated as an aimed "touch" (a *selection* rather than an *excitation* gesture [20]) since the performer must maintain contact with one point on the interface for a brief but significant amount of time.

Data from the piezoelectric sensor are used to determine whether the touch had signif-

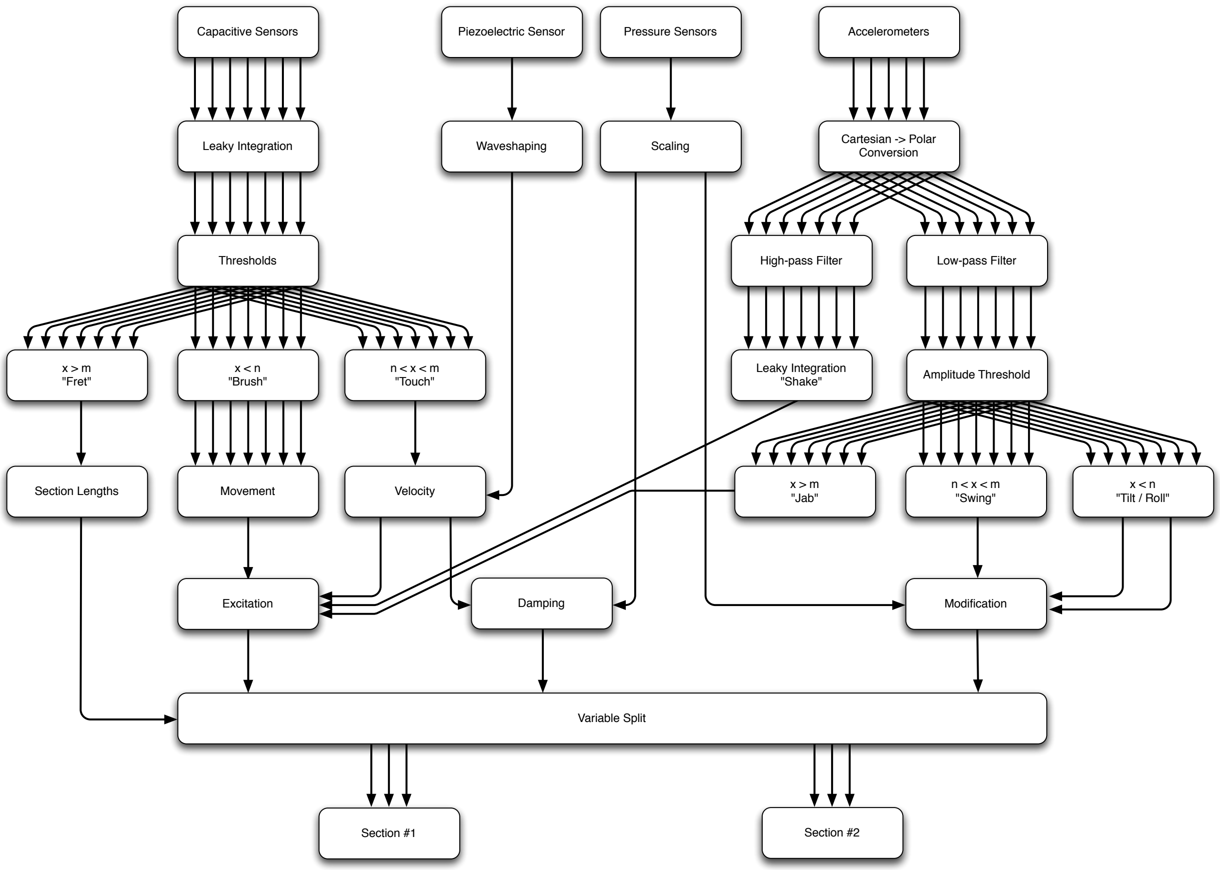

Fig. 4.3 A simplified diagram outlining the complex mapping relationships in the T-Stick software.

icant velocity (a strike or tap) and if so the newly detected touch is mapped to excitation. If no peak is detected in the piezoelectric sensor data, the information on the activated sensor's location is processed to calculate how close the position touched is to integer divisions of the active T-Stick section, and this information is mapped as location-specific instrument damping (see *Developing the Interaction Metaphor* below).

Lastly, if the integrated sensor data are greater than the second threshold the touch is interpreted as a "fret," dividing the length of the physical interface. In this way the division of the interface can be adjusted by the performer over time, changing the lengths of the open sections between grips and the ends of the physical interface. The section length data are used to determine in which section other excitation and damping gestures occur, and are also exposed as a parameter to the second mapping layer. At this point in the mapping process, lengths are measured in meters rather than number of capacitive sensors, allowing the sensing resolution to be improved in future models without compromising the remainder of the mapping system.

4.2.2 Accelerometers

The "cooked" accelerometer data arrive already converted to polar coordinates and with normalized amplitude. These data are passed through digital filters to split their low-frequency and high-frequency components. The high-frequency components of the accelerometer signal are considered to correspond to rapid shaking of the physical interface, and are passed on to become part of the excitation signal. Each end of the interface can be shaken somewhat independently in the x-y plane (zero elevation in polar coordinates) and so the information from each end of the interface is routed to "excite" the appropriate section. The low-frequency components are split according to another threshold on their amplitudes; low frequency, low amplitude signal is interpreted as tilting or rolling of the interface and is exposed for mapping as independent parameters, whereas signals with higher amplitude are interpreted as jabbing, swinging or spinning of the interface and contribute to the excitation signal in addition to being exposed as independent parameters.

4.2.3 Piezoelectric Sensors

The "cooked" signal from the piezoelectric sensor is further processed using waveshaping with a custom-designed transfer function in order to maximize useful signal. The resulting data are used to differentiate between strikes or taps on the interface and light "damping" touches as described above, and to indicate the velocity of percussive attacks. In addition to these uses, the output signal also peaks when the interface is twisted using two hands. The twisting gesture is exposed as a separate parameter for mapping, and is detected when the piezoelectric sensor signal exceeds a threshold while two detected hand-grips remain unchanged.

4.2.4 Pressure Sensors

Pressure sensor data, in addition to being available as a `/cooked` parameter, are also used to influence the level of the abstracted "damping" signal. In this role, the pressure sensors may be used to quickly silence a synthesizer

Table 4.2 Semantic gesture parameters, with n as the instance ID.

Namespace	Data Type	Units	Min	Max
/tstick/n/instrument/length	f	metres		
/tstick/n/instrument/length/left	f	normalized	0	1
/tstick/n/instrument/length/right	f	normalized	0	1
/tstick/n/instrument/jab/amplitude	f	normalized	0	1
/tstick/n/instrument/jab/angle	f	degrees	0	360
/tstick/n/instrument/jab/elevation	f	degrees	-90	90
/tstick/n/instrument/tilt	f	degrees	-90	90
/tstick/n/instrument/roll	f	degrees	-90	90
/tstick/n/instrument/twist	f	normalized	0	1
/tstick/n/instrument/squeeze/left	f	normalized	0	1
/tstick/n/instrument/squeeze/right	f	normalized	0	1
/tstick/n/instrument/excite/left	f	normalized	0	1
/tstick/n/instrument/excite/right	f	normalized	0	1
/tstick/n/instrument/damp/mode	f	normalized	2	6

4.3 Semantic Mapping

This layer is where the character of the musical interaction is determined. Mapping inputs to outputs in this layer is essentially arbitrary in that the data itself does not contain information about what it should do. Mapping at this level could be seen as a matter of personal taste or bias; however in the case of the T-Stick, a commitment exists to make the mapping reflect a kind of physically-informed interaction model, in which a user should be able to quickly construct a personal model of "how it works." This task is "not trivial"[38] and indeed comprised a very large part of software development effort and time. This section will describe the interaction design behind the T-Sticks, and also outline the design and implementation of the mapping system.

4.3.1 Developing the Interaction Metaphor

In keeping with the design goals stated in Chapter 1, interaction with the T-Stick is intended to be analogous to interaction with an physically vibrating object. It is not meant to be a strict physical model, however, but rather to make use of a performer's previous experience to guide the interaction. References to physical behavior can be warped or stylized.

Specifically, it was decided that each end of the T-Stick interface would metaphorically "make sound," and that the performer could excite sound in the two ends independently. Touch is mapped subtly to many different parameters, depending on where, how, and how long the interface was touched. Holding the interface breaks or frets the virtual vibrating object, and the frequency of possible vibration is based on the free length of the interface excited (between the end and the closest touch). Damping of sound is also controlled primarily by touch. All excitation of sound is mapped from gestures that require physical energy from the performer.

In order to maintain the interaction metaphor, and to distinguish between the members of the T-Stick family, the fundamental pitch of each DMI is based on its total length. Efforts are also being made to include differences in timbre, instrument response, natural decay-time, maximum volume, and effort required for excitation of sound.

4.3.2 The Mapping System

Initially, the T-Stick mapping software was envisioned as a closed, "black box" system, in which users of the interface would never be exposed to the OSC address namespace or parameter data. This system would be made up of all the elements of the software framework discussed here, but the individual parts would be hidden and the only output would be sound. Due to the demands and opportunities presented by the McGill Digital Orchestra project [42], in which composers, performers, and instrument designers work collaboratively in the development of DMIs, and which will feature the T-Stick consort in its final performances, the original plan was revised. In the longer term, technical and even semantic parameters will be hidden from the performer, but in the shorter term it was deemed useful to use the input of performers and composers in the development of specific mappings. To this end, a network-based "plug and play" mapping system was developed in collaboration with my colleague Stephen Sinclair. A more in-depth discussion of the mapping system implementation will soon be published [43]; the description here will be more brief. For more information about the McGill Digital Orchestra project as it pertains to this thesis, please refer to section 5.1.2.

Motivation

In the process of creating DMIs for the Digital Orchestra project, we found ourselves faced with the challenge of forming the semantic mapping relationships in collaboration with performers and composers tasked with playing and writing for the new instruments. Since the project tackles instrument-building, composition, and performance of DMIs simultaneously, we found ourselves in need of flexible, easy-to-use tools to help optimize the mapping process. We needed to be able to modify connections between data streams during precious engineer-composer-performer meeting time, while minimizing wasted minutes spent "reprogramming" the signal processing routines. Although arguably both powerful and intuitive, the graphical environment provided by Max/MSP did not seem appropriate for these purposes, since non-programmers with limited familiarity with such tools were expected to help in experimentation and design.

Network Environment

In consideration of several ongoing projects, including GDIF [41], Jamoma [44], Integra [45], and OpenSound Control [46], we created a "plug and play" network environment for controllers and synthesizers. Devices are able to announce their presence on the network and make their input and output parameters available for arbitrary connections, negotiated using a graphical user interface implemented in Max/MSP. Any controller is able to connect to any synthesizer "on the fly," while performing scaling, clipping, and other operations on the data.

Entities in the "orchestral neighbourhood" are classified as being *controllers*, which consist primarily of outputs, *synthesizers*, which consist primarily of inputs, or *routers*, tasked with performing address translation and data processing. Currently, controllers send their data streams to a router which performs address mapping and scaling before re-transmitting to a synthesizer.

Perhaps the most important function of the mapping system is automatic device and namespace discovery. As each device becomes available on the network, it negotiates with existing devices for a unique port number and ID, and then announces its presence with a message containing its name, IP, port, and device type (output, input, or router). This information can also be queried by another device or from the Graphical User Interface (GUI), along with the OSC addresses of the devices entire parameter-space. Each parameter

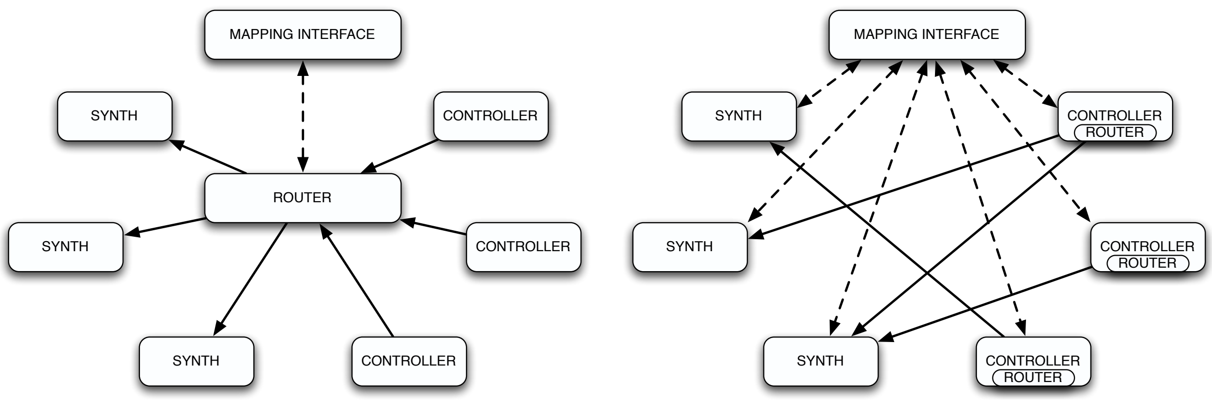

Fig. 4.4 The mapping system currently employs the centralized topology shown on the left. Future development will instead follow the distributed network model shown on the right.

address optionally has associated with it a data type (integer, floating-point, etc), a unit type (Hz, meters, etc) and minimum and maximum values. This information, if provided, facilitates the quick negotiation of linear scaling, unit-conversion, and automatic calibration of data streams.

Connection Processing

When a mapping connection is made, a router begins to listen for the mapped controller address pattern, and translates the OSC address string to the appropriate synthesizer parameter name. Before sending the data, the router usually performs scaling or other processing to match the ranges or units of the output and input parameters. The mapping system implemented allows the use of any mathematical expression understood by the Max `expr` object to process that data; this expression may be altered in real-time, causing the dynamic instantiation of a new `expr` object with the appropriate arguments.

One interesting data processing tool that we are exploring is a filter system for performing integration and differentiation. We often find during sessions that a particular gesture might be more interesting if we could map its energy or its rate of change instead of the value directly [47]. Currently the data processing is limited to first-order FIR and IIR filtering operations, and anything more complex must be added as needed to the "gesture" mapping layer and the data streams included in the mappable namespace.

GUI

A mapping interface was developed to act as a graphical user interface (GUI) for mapping tasks. It forms a separate program from the Router, and communicates with controllers, synths, and routers using OSC. In addition to allowing the negotiation of mapping connections from another location on the network, this approach allows for the simultaneous use of multiple mapping interfaces, with multiple users collaborating to map the parameters of a common set of controllers and synthesizers. When a mapping interface is launched, it queries discovered routers for their connection status, and connections are displayed on the screen for selected devices. The mapping interface has several main functions:

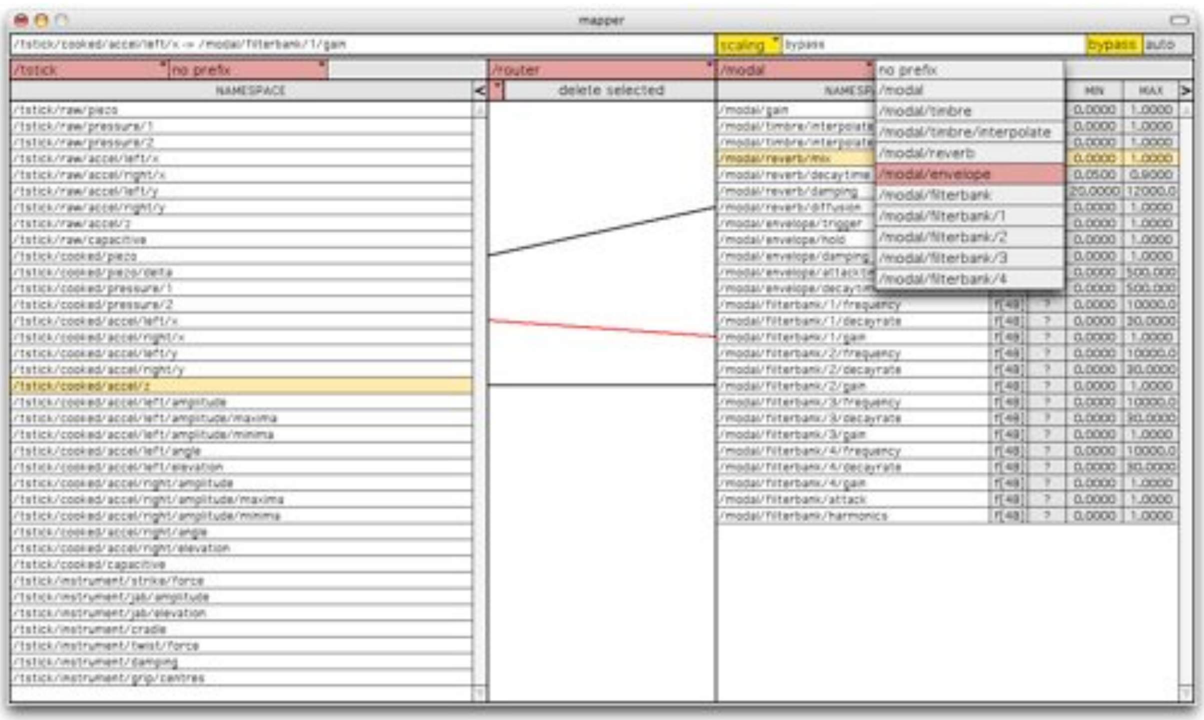

Fig. 4.5 A screen shot of the Mapping Interface.

Loading and saving mapping-sets. Mapping connections, including the applied signal processing equations and settings, can be saved to disk and loaded for later use or improvement. This task, though negotiated using the mapping interface, is actually handled by the router holding the connection information.

Browsing the Network Neighbourhood. Another use of the mapping interface is naturally choosing the devices you wish to work with, both for gesture and for sound synthesis. The interface queries devices on the network, discovering whether they are devices with mappable inputs, outputs, or both, and displaying this information to the user in an easily understandable way. Simple drop-down menus are used to select a device or devices.

Browsing and searching namespaces. In this capacity the mapping interface communicates directly with the various devices present on the network, requesting them to report their namespaces when necessary and displaying this information to the user. In order to manage the browsing and mapping of very large namespaces, the mapping interface also features searching using pattern-matching. Two stages of namespace filtering are available, which may be used together. One stage allows filtering by OSC address-pattern prefix, chosen from a drop down menu, so that the user may view the set of parameters which are children of a particular node in the address hierarchy. The other stage allows filtering by regular expression, so that only parameters matching a particular pattern are displayed.

Negotiating mapping connections and properties. In this capacity the mapping interface communicates only with a specific router, acting as a graphic UI for controlling the router's function, and supplying important visual feedback regarding the router's internal state to the user. Simple and intuitive methods are provided for creating and destroying mapping connections, and for editing the properties of existing connections (ie: scaling, clipping). Connections are created simply by selecting displayed namespaces on each side of the interface (inputs and outputs), and lines are drawn connecting mapped parameters. Mapping connections can be selected (in which case they are displayed in red) for editing or deletion. By selecting multiple namespaces or connections, many mappings can be created, edited, or destroyed together.

When a connection is made, the mapping tool defaults to performing no operation on the data ("bypass"). A button labeled "auto" instructs the tool to perform basic linear scaling between the provided data ranges. A toggle switch turns on and off calibration of the scaling using the detected minima and maxima of the input data stream. The user can also manually type "auto" in the expression textbox with arguments defining a specific input and output range. Options are also available for defining a clipping range. An expression box is also provided for specifying arbitrary functions for scaling the data.

Usage and Utility

From their earliest use, the mapping solutions we developed allowed us to streamline the process of mapping in collaboration with performers and composers. The ability to quickly experiment with a variety of mapping connections saves time, and thus democratizes the mapping process in that it is possible to try *everyone's* ideas during a mapping session. Showing the performers that the connections are malleable in type and processing allows them to focus on optimizing gestures and makes them less likely to just accept the status quo. Composers are able to develop mapping-sets between group meetings (and without the supervision or assistance of a technical member), and then demonstrate their progress to the group using the saved mapping file. Using a common platform and GUI for the group also saves time and effort, since the work of others is easily viewed and understood.

Controllers and synths that are still in development are also easily supported: as the supported parameter-space increases, the device simply presents more namespaces to the GUI.

Naturally this system does not solve all of the problems encountered in a collaborative effort of this type. The technical knowledge of the group members varies widely, and some technical knowledge of the individual controllers and synths is still necessary, not least because they are still in development and may not always respond predictably. As much as possible, however, we have made the connection, processing, and communication of data between devices easy to both comprehend and perform.

4.4 Musical/Sound Semantics and Third Mapping Layer

This mapping layer makes abstracted, higher-level musical parameters of the synthesizer available as mappable controls for the user, and performs "technical" mapping to the underlying synthesizer inputs. Rather than dealing with low-level parameters like frequency, gain, or modulation indexes, this layer takes more abstract musical parameters as input, such as pitch, timbre, and dynamics. The parameters available here may still depend somewhat on the particular synthesis method chosen: for example, not all synthesizers are designed to allow control over pitch. For the most part however, mapping connections made to semantic sound parameters should be transferable between different synthesis algorithms.

A common mid-level musical control available in many commercial MIDI synthesizers is the application of *envelopes* to control time-varying parameters. Historically, this has been used to compensate for the lack of continuous control available to users of commercial keyboard interfaces, and has been used to create more realistic volume and timbre envelopes for non-percussive sounds (winds, brass, bowed strings) than would otherwise be possible. For the most part, a DMI designer may wish to avoid envelopes, and instead opt for continuous control over all parameters by the performer [10]. It should be noted, however, that envelopes can still have a place even in contexts in which designers wish to maximize continuous performer control, at least in a simple form in which they simulate the natural acoustic decay of a sound.

As before, the technical mapping between semantic and technical synthesis parameters is incorporated into the synthesizer software patch.

Table 4.3 Semantic synthesis parameters, with n as the instance ID.

Namespace	Data Type	Units	Min	Max
/synth/n/note	f	midinote	0	127
/synth/n/envelope/trigger	f	normalized	0	1
/synth/n/envelope/hold	f	normalized	0	1
/synth/n/envelope/damping	f	normalized	0	1
/synth/n/envelope/attacktime	f	ms	0	500
/synth/n/envelope/decaytime	f	ms	0	1
/synth/n/timbre/density	f	normalized	0	1
/synth/n/timbre/smoothness	f	normalized	0	1
/synth/n/timbre/harmonicity	f	normalized	0	1
/synth/n/damping/harmonic	if	na/normalized	0	10/1

4.5 Synthesis Parameters

The last layer is responsible for actually generating audio. Its inputs are specific to the synthesis method used, and are sometimes highly technical.

During development of the T-Stick family, several approaches have been taken to synthesizing their voices, several of which have been featured in T-Stick performances. One of the first sound-synthesis attempts for the T-Stick was implemented by D. Andrew Stewart using Max/MSP. Stewart explains the approach:

> This initial attempt at designing a synthesis engine for the T-stick involves both audio file manipulation in real-time and physical modeling synthesis. Generally speaking, the manipulation of audio samples is responsible for the sustained portion of the instruments voice, while the more transient qualities of the T-stick are a result of physical modeling. [...] The final versions of the engine use audio files of heavy metal music, ABS pipe sound, a carillon bell, metallic sounds from a sample of John Cages First Construction in Metal, a sinusoid and sawtooth waveform. [...] All of the audio passes through the `munger~` object (PeRColate version 0.9b5), using a moderate to large grain size. The effect is a sort of smearing of onset qualities, instead of a granular timbre. [...] The physical modeling aspect of the T-sticks voice is realized using Perry Cooks `ublotar~` object, which offers a flute/electric guitar model. [48]

4.5.1 The *Modal4* Synthesizer

The main software synthesizer constructed for the T-Sticks uses the *modal synthesis* technique, which involves calculating a series of filter coefficients from sound-file analysis or a physical model, and using this information to control a bank of resonant filters [49]. The synthesizer is implemented in Max/MSP, using CNMAT's `resonators~` object to supply a bank of resonant filters [50]. Settings for the centre frequency, gain, and decay time for each filter can be loaded from presets, read from an analysis file, or set *en masse* by drawing in a graphical window supplied.

The entire software synthesizer is built to be controllable either with a built-in GUI or via OSC, and is compatible with the mapping system described above. Excitation of the filter-bank is performed using pink noise passed through a flexible enveloping function, built to emulate the behaviour of a vibrating object which can be damped. Controls over attack and decay times, impulse timing and strength, and a damping coefficient are all available via OSC. The entire control namespace for the *Modal4* synthesizer is shown in table 4.4.

To protect audio hardware from the extremely low frequencies which can be produced by this synthesizer, a high-pass filter is applied to the output. A reverb effect is also incorporated into the synthesizer to add basic room effects and optionally place the sound in a subjectively more natural sonic space.

Table 4.4 Technical synthesis parameters, with n as the instance ID.

Namespace	Data Type	Units	Min	Max
/modal/n/frequency	f	Hz	20	10000
/modal/n/note	f	midinote	0	127
/modal/n/gain	f	normalized	0	1
/modal/n/timbre/interpolate/x	f	normalized	0	1
/modal/n/timbre/interpolate/y	f	normalized	0	1
/modal/n/envelope/trigger	f	normalized	0	1
/modal/n/envelope/hold	f	normalized	0	1
/modal/n/envelope/damping	f	normalized	0	1
/modal/n/envelope/attacktime	f	ms	0	500
/modal/n/envelope/decaytime	f	ms	0	500
/modal/n/filterbank/n/load	s	na	na	na
/modal/n/filterbank/n/frequencies	f[48]	Hz	20	10000
/modal/n/filterbank/n/decayrates	f[48]	n/a	0	30
/modal/n/filterbank/n/gains	f[48]	normalized	0	1
/modal/n/filterbank/n/basefrequency	f	Hz	20	10000
/modal/n/filterbank/n/basenote	f	midinote	0	127
/modal/n/filterbank/n/frequency	f	Hz	20	10000
/modal/n/filterbank/n/note	f	midinote	0	127
/modal/n/filterbank/n/transpose	i	n/a	0	1
/modal/n/filterbank/attack/gains	f[48]	normalized	0	1
/modal/n/filterbank/damping/harmonic	if	na/normalized	0	10/1
/modal/n/filterbank/n/filter/n/frequency	f	Hz	20	10000
/modal/n/filterbank/n/filter/n/decayrate	f	n/a	0	30
/modal/n/filterbank/n/filter/n/gain	f	normalized	0	1
/modal/n/reverb/mix	f	normalized	0	1
/modal/n/reverb/decaytime	f	s	0.05	0.9
/modal/n/reverb/damping	f	ms	20	12000
/modal/n/reverb/diffusion	f	normalized	0	1

In addition to the usual functionality and timbral flexibility of modal synthesizers, this particular implementation of modal synthesis differs from others in two important aspects.

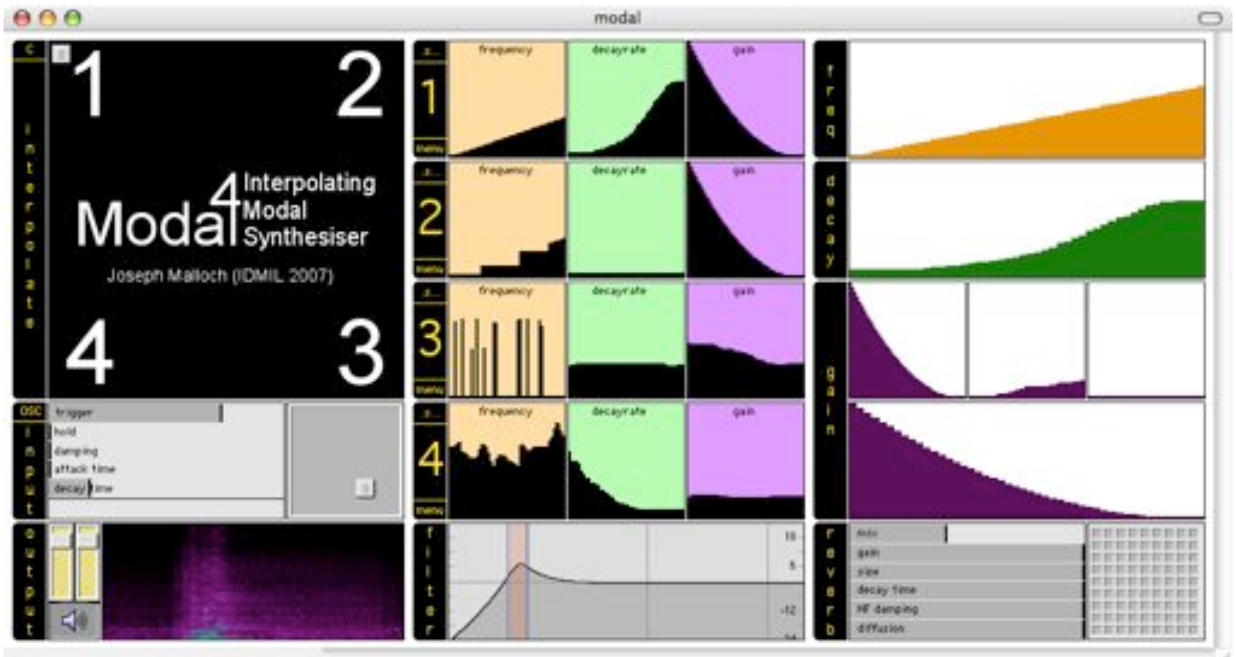

Fig. 4.6 A screen shot of the software modal synthesizer built for the T-Stick.

Interpolation

The *Modal4* synthesizer features four "presets" of filter-bank coefficients, with mechanisms provided for reading and writing preset files, copying parameters from one preset group to another, and modifying existing coefficients at will. Although the text storage format used for saving filter coefficients separates frequency, gain, and decay values into separate lists, preset modules can also load sound-file analyses in triplet form (as used natively by the `resonators~` object) and automatically demultiplex and reformat the coefficient data.

The user can switch quickly between these presets, potentially changing the timbre dramatically, or interpolate between the settings from the four loaded presets using list-interpolation. This allows gradual, more subtle mixing between the respective timbres of the presets. Interpolation is controlled either by using a two-dimensional graphical interface object or by using the `/timbre/interpolate` OSC message.

A modal synthesis preset does not necessarily have an associated pitch, and even when it does, the fundamental pitch does not always correspond to the frequency of the lowest-sounding filter. When using pitched presets, interpolation can cause distracting pitch-sliding effects in addition to timbral evolution. To avoid this situation, each preset can be assigned a base note or frequency, and will perform variable scaling to individually match the globally-set note or frequency. This allows exploration of the timbre space without bending the pitch. Transposition of each preset is optional, and can be disabled if the user does not wish a particular preset to be transposed.

A fifth filter gain preset is provided and connected to the output of the main envelope control, allowing a user-definable *attack spectrum* which varies with the gain envelope of the sound. In practice, this is usually used to insert noisier high frequencies into the output during the beginning phases of percussive attacks and during continuous excitation corresponding to bowing.

Harmonic damping

A second powerful feature of the *Modal4* synthesizer is the ability to perform selective damping of spectral harmonics. Intended to be mapped to an integrated version of the touch-sensor data, this feature allows for "natural" frequency-damping without using a full physical model of a vibrating string or bar.

For simplicity, the harmonic damping function assumes an ordered harmonic relationship between filter frequencies, in which the first filter is tuned to the fundamental frequency and subsequent filters are multiples of this frequency. The OSC message `/modal/n/filterbank/damping/harmonic 2 0.5` will apply a normalized damping factor of 0.5 to a virtual "node" dividing a virtual vibrating string into 2 equal parts. When applied to a harmonic-series modal preset, this effectively results in the damping of all odd harmonics, a fundamental one octave higher, and a thinner timbre. This process is clarified somewhat in figure 4.7.

If harmonic damping is applied to modal presets not arranged in a harmonic series, the pitch and timbral changes will not be as easily predicted, but may nevertheless result in interesting sounds.

4.6 Data-logging & Playback

This area of the patch is not a mapping layer, but rather a means to record the gestures of a performer and then replay them through the mapping layers for refining and adapting the mapping. This allows "off-line" editing of the patch when the performer is not present.

4.7 The Digital Orchestra Toolbox

In the process of creating patches for this thesis project and for the Digital Orchestra project (see section 5.1.2), an effort was made to modularize any commonly used subroutines. These have been organized, with help patches, into a toolbox of useful Max/MSP abstractions. The "Digital Orchestra Toolbox," or *dot*, presently contains over 40 different abstractions with help files, of which this author directly contributed 30. Functions developed specifically for the T-Stick and the *Modal4* synthesizer were added to this collection, which will soon be freely available on the Digital Orchestra website [42].

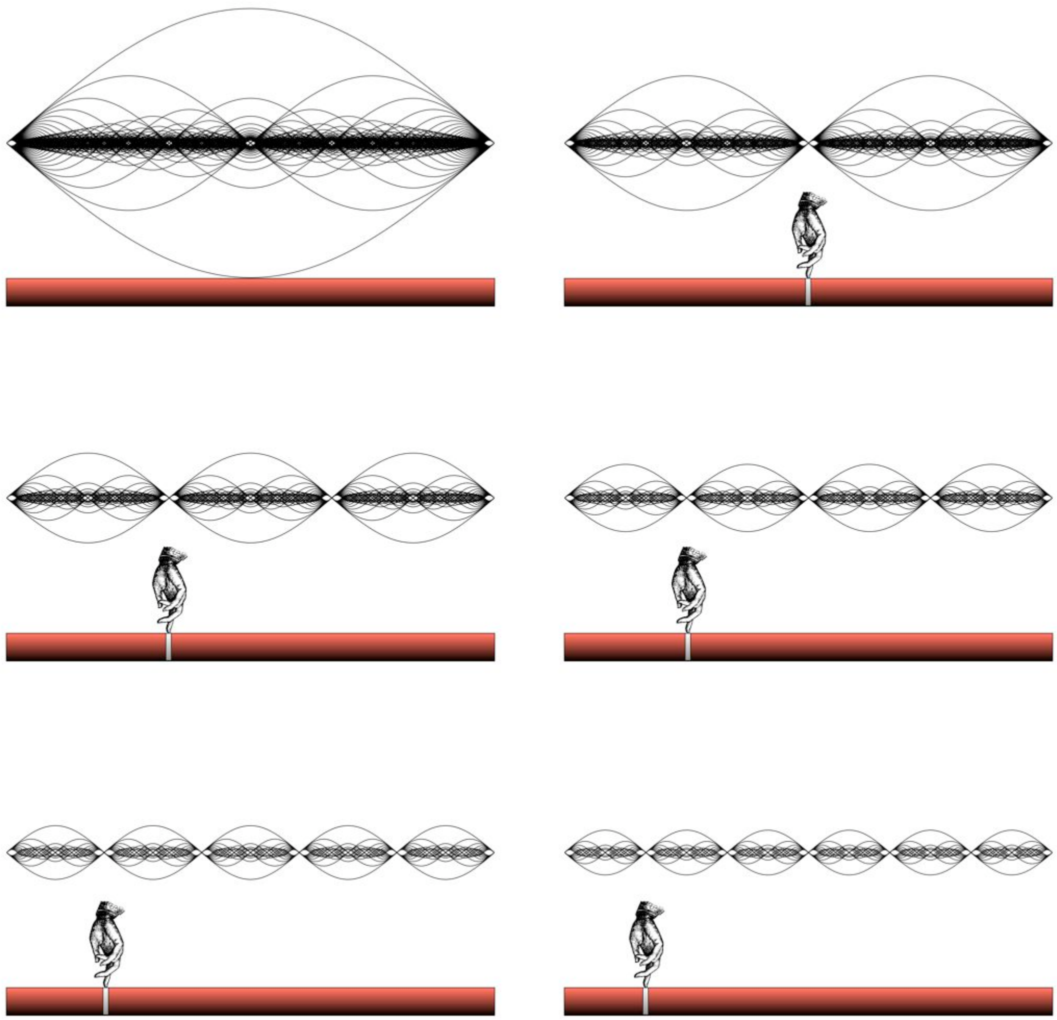

Fig. 4.7 Diagram showing the effect of playing harmonics on the T-Stick. For each position touched, virtual vibration modes are damped if they do not have a node at that location, and are filtered in modal synthesis.

Chapter 5

Musical Performance

The last two chapters covered the technical and implementation issues encountered during this thesis project in depth. It should not be forgotten, however, that the end goal of the project is musical performance. The instrument can only truly be tested though the process of rehearsing and performing music. Cook's assertion that DMI designers should "make a piece, not an instrument or controller" [17] was intended to refocus the process on musical performance. Without this focus the musical results of a project such as this tend to be flat and unarticulated [51]. Nevertheless, it is not uncommon for DMI projects to become overwhelmed by the technical requirements and never reach their aesthetic potential. A primary goal in this thesis project was to avoid this pitfall, and ensure that musical performance remained at the center of attention.

5.1 Performer Collaboration

Much of the T-Sticks' development took place in collaboration with performers in the context of two organized sessions: a seminar taught at McGill University, and as part of a larger effort entitled the *McGill Digital Orchestra Project* [42]. Many design decisions were made in consultation with performers, and their feedback throughout the development process was invaluable. In the following sections, I will briefly outline the collaborative process and the performers' contributions within these two contexts.

5.1.1 Digital Musical Instruments Seminar

Approximately every two years an interdisciplinary seminar on Digital Musical Instruments is offered at The Schulich School of Music. In contrast to seminars on HCI and sensing offered as part of the Music Technology program, this seminar is offered equally to performers, composers, and music technology students, and is taught jointly by Marcelo Wanderley from the Input Devices and Music Interaction Lab (IDMIL) and Sean Ferguson, who heads the Digital Composition Studio (DCS). Students are asked to form groups in which each discipline is represented, and are tasked with designing, constructing, composing for, and performing on a new digital musical instrument over the course of one semester. This extremely accelerated schedule has nevertheless resulted in some very interesting projects, from both technological and artistic perspectives. Video recordings of some of the past seminar performances are available on the DCS website [52].

The first T-Stick prototype was constructed in the context of this seminar, and as such had the benefit of early involvement and testing by two doctoral-level performance students, and the added push of a concert deadline at which two short original pieces composed by other members of the group were played. The main goals, instrument shape, and sensing methods had already been finalized before the seminar began, but performers participated in many of the remaining design decisions:

- **Interface lengths.** It had already been proposed that the T-Stick be built in several sizes, so as to form a consort of DMIs, but the starting length had not been finalized. Through discussing possible gesture vocabulary, and experimenting with different lengths of ABS pipe, the performers helped to determine the final length of the first prototype. Lengths of subsequent versions are intended to contrast with this first prototype (see table 3.1).

- **Spacing of capacitive sensors.** The capacitive sensing surface of the T-Stick is intended as a multi-touch position sensor, however time constraints dictated that the resolution of the first prototypes be limited. Since the interface was called upon to sense the difference between the touch of a finger and that of a hand, it was decided to use a minimal sensing resolution corresponding to a natural grip on the surface, such that one sensor would be covered by each finger. Each member of the group was asked to test the spacing, and evaluate it for comfort.

- **Length and location of pressure sensors.** Initial plans included a single long pressure sensor covering the entire length of the interface. Performer comments led to the current configuration, in which there is space in the middle and at the ends of the interface at which it can be held without pressing on the pressure sensors.

- **Foam thickness over pressure sensors.** Many pressure sensors – including commercial FSRs and the custom paper sensors used in this project – are *isometric* sensors, meaning that they sense change in pressure without a change in position [53]. In everyday life, however, we usually only think to deliberately apply pressure to objects that move or deform from the applied pressure. To give a more natural “feel” to the pressure sensors, and also provide added feedback to the performers, the sensors were covered in closed-cell foam. Performers were consulted on the type and thickness of foam to use.

- **Frets.** The first prototype of the T-Stick features frets between each of the capacitive touch sensors (see section 5.3 for discussion on the frets). Early in development, a variety of different materials were used to form frets on a mock-up of the T-Stick and covered with shrink-tubing to simulate the feel of a finished interface. A consensus was reached within the group on the “best-feeling” fret, which was incorporated into the final version. Additionally, at the performers’ request coloured fret-markers were added at sixth-divisions of the interface. These were intended for quickly finding a particular position, similarly to the way coloured strings are used for pitch-classes C and F on a harp.

The entire group (performers, composers, and music technology students) participated in discussions regarding the “voice” of the DMI. It should be noted, however, that these discussions impacted the original mapping and synthesis algorithms, which have since been replaced by the systems described in Chapter 4.

The seminar concert supplied the first real test for the T-Stick, since preparation required daily practice with the interface by the two performers in the group. The goals of robustness and simplicity seemed to be met quite well, since the prototype was used approximately six hours every day for more than two weeks with only one repair, and the performers learned quickly to operate the hardware and software components of the instrument without technical supervision. Both performers commented that they did not believe playing the T-Stick made use of their existing specialized motor skills (as a percussionist and a pianist, respectively), but that the as-yet-unrealized potential of the T-Stick for performance was exciting. Xenia Pestova (the pianist) wrote "The design is slick, sturdy and appealing and the interface is easy to handle (at the same time demanding practice and dedication to discover the finer and subtle aspects of control, which is a rewarding combination for the performer)" [48].

The bulk of the problems encountered by the performers in the context of the seminar were attributable to the lack of sufficient time to become comfortable with the new instrument. Fernando Rocha (the percussionist) wished that the process could be more like that of learning a traditional acoustic instrument, in which one starts with fundamentals, such as long notes, scales, and arpeggios: "Since in the seminar we had to learn the instrument at the same time as learning a piece, I felt as [if] I was missing important phases of the development of my technique (almost the same as starting to play a clarinet learning a piece based on extended techniques)" [48].

5.1.2 McGill Digital Orchestra

The McGill Digital Orchestra is a research/creation project supported by the Appui la recherche-création program of the Fonds de recherche sur la société et la culture (FQRSC) of the Quebec government. The project brings together research-creators and researchers in performance, composition and music technology to work collaboratively in creating tools for live performance with digital technology [42]. A large part of this research focusses on developing new musical interfaces. The grant has a duration of three years and will culminate with a performance of new works during the 2008 MusiMars/MusiMarch Festival in Montréal, in at least one of which the T-Sticks will form part of the ensemble.

In the context of the project, student research-assistants were hired from the three contributing academic areas; currently this means three doctoral students in performance work regularly with the T-Stick as part of their research contribution to the project. Thus far in the project, performer collaboration has been focussed on extending and refining the mapping relationships used to translate performer gesture into sound. To this end, weekly meeting/mapping sessions are held in which group members collaboratively develop instrument voices and playing techniques. The mapping system described in Chapter 4 was developed to make these meetings more productive and enjoyable, and using these tools the “new and improved” mapping and synthesis for the T-Stick are taking shape.

Two of the performers involved in the Digital Orchestra project were also team members in the seminar project described above, meaning they have now been working with the T-Stick DMI for one year. During that time, several other performances have taken place, detailed in table 5.1. Of particular note, Fernando Rocha chose to perform a piece written for T-Stick on a graded lecture recital for which he committed to running all electronics himself with no technical assistance. A DMI not designed with robustness and simplicity in mind would probably not have permitted this freedom.

5.2 Technique

Early in the project several possible playing techniques were identified and described, of which two were further developed through collaboration with performers and the composition of works written for specific performers and performance modes. The T-Stick is played in the first mode by holding the interface in front of the body with both hands, and manipulating it in open air (see figure 5.2). The second mode involves the use of an adjustable spike which is fastened to the T-Stick and rested on the floor in the style of a ’cello or bass (see figure 5.1). This technique has the advantage of freeing a hand; however keeping the interface in contact with the floor obviously restricts the possible gestures in other ways. Naturally a performer is free to change between techniques, but as the spike adds considerable weight to the interface, performers generally choose to remove it if they are planning to perform using the first mode.

The performance gestures using the T-Stick are closely related to the mapping implementation discussed in Chapter 4, so a brief description here should suffice. A wide variety of possible gestures and postures are displayed in figure 5.2.

- **Excitation.** As planned in the project goals, excitation of sound is possible through a number of techniques. Quick attacks are possible by striking the interface with the hand; sustained sounds with slower attacks are accessed by brushing the touch-sensitive surface, shaking the interface rapidly, or swinging it through space.

- **Damping.** The performer damps a sound by lightly touching the interface (a “heavy” touch becomes a percussive strike). A light touch that is quickly released selectively damps overtones but lets the remaining sounds continue to ring (with their “natural”

Fig. 5.1 Xenia Pestova playing the T-Stick in 'cello position during a concert.

decay), and a wide touch damps most or all of the sound. Squeezing the interface is also mapped to damping and can be used to access very short, muffled sounds, however this technique has not yet been fully exploited due to problems with the original pressure sensors.

- **Selection.** The performer can select which part of the interface to play and "fret" the touch-sensitive surface, thus changing the pitch of the resulting sound. Also, since each end of the T-Stick is mapped separately to sound-synthesis, the performer can choose which end to play or can excite sound in both and play two sounds simultaneously.

- **Modification.** Modification of the volume of sound is possible only through the excitation techniques. Modification of the timbre is performed by manipulating the orientation of the interface: tilting, rolling, and rotating, usually in combination with excitation techniques.

The performers worked very hard to create a vocabulary of gestures idiomatic to the interface. They lost no time in discovering and making use of new playing techniques:

subtle twisting of the gestural controller (which deforms the piezoelectric disc intended for sensing velocity of bumps and hits) has now become a “standard” part of T-Stick performance practice and pedagogy.

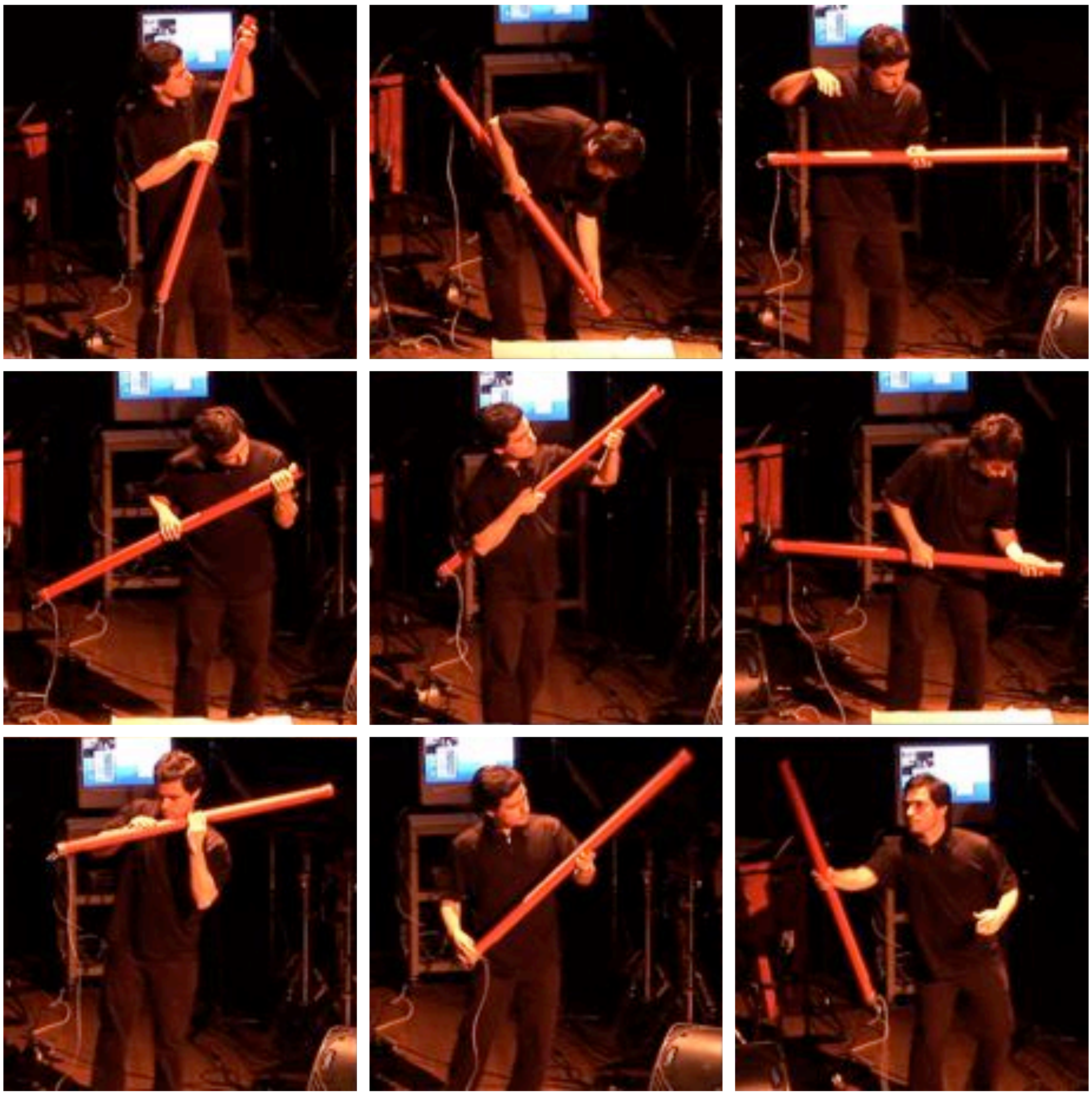

Fig. 5.2 Fernando Rocha performing the T-Stick in concert. Note the different playing technique compared to figure 5.1.

5.2.1 Effort, Timing, and Expression

Performance with the T-Stick is still at a very early stage compared to most traditional instruments: only three pieces of music have been composed for it, and there are currently only a few people who play it, and not one who has played it for longer than a year. Nevertheless, they are accomplished performers on their own instruments, and contribute their musicality and years of stage experience to the process of learning the T-Stick. Subjectively, Rocha's performance after playing for six months was vastly more musically satisfying than the performances after one month; hopefully an indication that the T-Stick is becoming an instrument through which musical ideas and emotions can be successfully expressed.

Most early frustrations were related to the mapping of gesture to sound excitation. Preliminary parameter mapping linked touching the interface to triggered sounds, a situation which led performers to complain that they had trouble *not* making a sound with the interface, and thus did not feel they were truly in control of the instrument. This was not surprising, since the original project goals stated that effort should be required to produce sound, but the performer comments helped cement the effort-excitation relationship as a primary concern in mapping the T-Stick. Also related to effort, Pestova commented that the first prototype was perhaps too light for performing with the spike, since it lacked the weight of a 'cello or bass clarinet and thus tended to slide on the floor. It is probably not desirable to increase the weight of the interface, since it was generally considered comfortable for the open-air performance technique. In the future, the possibility of performing with a spike will likely be restricted to the heavier bass version of the T-Stick.

On another note, mapping involving effort should also carefully consider the likelihood of performer injury. The piece played by Pestova in the first concert relied heavily on the twisting technique described above (the first performer-developed "extended technique"), and she reported that her hands were often sore after rehearsals due to the strain of performing this technique for extended periods of time.

Another initial barrier to expressive performance was the latency and jitter of the sensor signals in the first prototype. Composers and performers alike wished to perform rapid percussive passages on the T-Stick, and quickly reached the limits of the system. Subsequent versions of the interface feature much faster communication of sensor data, vastly increasing the responsiveness of the DMI.

Table 5.1 Performances and planned performances featuring the T-Stick. In addition to these formal performances, the T-Sticks have been demonstrated informally many times.

Date	Event	Location	Performer	Piece
24/04/2006	Gestural Controller Seminar Concert	Thomson House McGill University	Fernando Rocha	“Dancing with a tiger” D. Andrew Stewart
24/04/2006	Gestural Controller Seminar Concert	Thomson House McGill University	Xenia Pestova	“Study no.1 for T-Stick” Aaron Lindh
09/05/2006	MiniMusic presentation	Tanna Schulich Recital Hall, McGill University	Fernando Rocha	“Dancing with a tiger” (excerpt) D. Andrew Stewart
24/11/2006	FQRSC site visit	CIRMMT	D. Andrew Stewart	étude D. Andrew Stewart
19/11/2006	Doctoral Lecture Recital	Tanna Schulich Recital Hall, McGill University	Fernando Rocha	“The One, a source of electrical & musical energy” D. Andrew Stewart
24/02/2007	Roots and Rhizomes percussion conference	Music Department, University of California, San Diego	Fernando Rocha	demonstration
spring 2008	MusiMars/MusiMarch Festival	McGill University	TBA	TBA

5.3 Discussion: Don't Fret

As described in Chapter 3, the first T-Stick prototype featured small fret-like protrusions, between the positions of capacitive sensors, to allow tactile navigation of the sensing surface. Much care and deliberation was taken over the "feel" of these protrusions, and many different materials were tested. Some frets were marked with colours for quick visual navigation. Finally, a type of cotton-wrapped steel wire intended for jewelry-making was determined to have the right feel through the shrink-tubing cover, allowing easy tactile navigation without preventing the use of smooth brushing gestures.

The other T-Sticks do not feature these frets, however, as it was deemed that their presence encouraged users to perceive the capacitive-sensing surface as 48 discrete touch sensors rather than a low-resolution multi-touch position sensor. As discrete touch sensors, each capacitive sensor is not very well suited to musical control, especially as they lack independent velocity sensing - as a single multi-touch sensor, however, mapping becomes much more interesting and performance potentially more expressive. Without the frets, the location of the copper strips is hidden from the user by the covering of shrink-tubing.

It is hoped that removing the temptation of knowing the sensor positions will benefit future mapping implementations, as well as encourage the performers to think of the interface as a single monolithic object rather than a collection of sensors. It may seem like a small issue, but considering that the intended interaction metaphor is dependent on intrinsically integral sensor data (in which many sensors are attached to one simple object), using the 48 capacitive sensors separably is entirely opposite to the design intentions.

An interesting dilemma arises: can the performers learn to forget their habit of viewing the multi-touch sensor as a sort of slow, non-velocity-sensitive keyboard, or are fresh performers required for studying interaction metaphor with the T-Stick?

Chapter 6

Conclusions and Future Work

This chapter summarizes the work presented in this thesis, offers some conclusions, and indicates the directions this research will take in the future.

6.1 Conclusions

This thesis began by exploring issues relevant to the classification, design and performance of digital musical instruments. Traditional acoustic and electric musical instruments are commonly grouped conceptually into families, based on timbre, sound production method, physical medium, or user interface. These sorts of classification formalize similarities and differences between instruments and are also used to make important performance practice, pedagogical, and compositional decisions; however digital instruments are often ignored in classification schemes. DMIs are typically unique in user interface and sound-production, a situation which has hindered development of performance practice norms and thus any coherent approach to ensemble performance. A family of DMIs would allow experimentation with composition and performance within existing ensemble idioms while using digital instruments.

The thesis project was motivated not only by this situation, but also by the idea of creating user interface objects which a performer could explore guided by a physically-informed interaction metaphor. By trying to make a system which emulates an acoustically vibrating object, with internally consistent rules of "physics" in which excitation, modification, and damping were mapped to natural-seeming gestures, it was hoped that both novice and expert users would find the interaction more amenable to musical expression.

Each member of the DMI family consists of hardware and software elements. The hardware "gestural controller" is designed and constructed to be extremely robust and simple to set up. Wherever possible, the sensors and circuits are hidden from the performer, not because they would not understand them, but because the technology becomes a distraction and creates the impression of a collection of parts rather than a monolithic view of the interface as a musical instrument. Each of the physical interfaces contains a completely multi-touch position sensor, two three-axis accelerometers, two pressure sensors, and a piezoelectric contact microphone. By processing and combining these sensor signals, the interface can sense touching, tapping, knocking, grasping, squeezing, brushing, shaking, swinging, tilting, rolling, and twisting gestures.

Wherever possible, data is transmitted using OpenSound Control with semantically-labeled address strings. A software mapping system was built to facilitate collaborative mapping sessions in which performers, composers, and technicians can define and refine mapping sets. This system is currently being used to collaboratively refine the programmed response of the T-Stick and its synthesized voice.

Some of the goals set at the beginning of the project have been surpassed, though they are generally the least spectacular. The interface design has proven to be extremely robust and the initial prototype is still working in near-perfect condition having required only a few small repairs over the course of the year. The T-Stick has also proven to be extremely simple to set-up technically - performers with very little technical background have been able to set it up and rehearse or perform with only a laptop and a USB cable. Technical help is only required in the rare event that something breaks, which occurs significantly less often than for some traditional musical instruments.

Meeting other goals will not be possible until a consort of T-Sticks is performed together in the near future. The remaining family members will soon be complete, and a piece is scheduled in early 2008 featuring a T-Stick quartet. The next steps in the development of the T-Stick are described in the following section.

6.2 Future Work

The T-Sticks are gaining maturity as DMIs, in that they have already been played for hundreds of hours in the lab, practice room and on the concert stage, but there are many changes and improvements to be made. On the purely technical front, iterative improvements to sampling rates, data throughput and position-sensing resolution will likely improve the performer-experience greatly, without dramatically affecting the techniques they use. More profoundly, the completion of the entire quartet of T-Sticks promises insight into the portability of skill between members of the family. Also, the degree of change that can be made to one T-Stick before the perception of family resemblance is “broken” will be explored. Is it enough to change one characteristic (i.e. size, shape, sensing, sound)? The challenge becomes to map and give voice to the members of the T-Stick family in such a way that they are perceived by performer and audience as complementary musical instruments rather than as redundant copies with superficial differences. This section outlines future work planned around the T-Stick project.

6.2.1 Sensing

Much of the planned future work on T-Stick sensing concerns the capacitive touch sensors. As discussed in Chapter 3, the resolution of touch sensing has been doubled since the first two prototypes were built, in order to allow the performer much finer control over excitation, damping, and pitch. This new resolution of sensing is not the end goal, however, since the original design calls for *continuous* multi-touch position sensing. Discrete sensing will probably continue to be used, but the resolution should be increased to the point where the performer cannot tell that the sensing is not actually continuous. This is achievable by using interpolation of sensor data in combination with higher-resolution sensing – perhaps using key electrodes printed on a flexible substrate and wrapped around the T-Stick. Alternatively, the use of continuous capacitive sensing could be explored, however this approach introduces many complications even as it solves others.

The use of multiple piezoelectric transducers in place of the present approach could potentially offer much better percussive attack information. Fairly simple algorithms which can run on ordinary microprocessors are available for calculating differential time-of-arrival of low-frequency acoustic energy arriving at separated microphones [54]. This would make a “tap location” parameter available for mapping, in addition to amplitude.

The addition of secondary sensing techniques, perhaps to differentiate T-Stick members, has also been discussed. Each type of T-Stick could possess unique abilities in addition to the base sensing described in Chapter 3. Suggested sensors include strain gauges, ultrasound distance-sensing (for sensing the distance to a wall or the floor), an air microphone, or a breath pressure sensor. The last option, in particular, would offer an additional excitation technique and could lead to some interesting didjeridoo-like mappings.

6.2.2 Data Processing

As mentioned above, efforts to increase sample rates and system responsiveness will continue. Beyond this, there is a commitment to gradually move more of the signal processing off of the laptop computer and onto the interface. In an ideal situation, if the interface is not an appropriate size or shape for embedding speakers, the interface should communicate all parameters as OSC messages. In addition, the interface should be made directly compatible with the mapping system described in Chapter 4, rather than requiring a software mediator. This should include the ability to announce its presence on the network and supply its namespace when requested. This improvement would truly push forward the goal of "plug and play" gestural controllers. Moving directly to network communication in the future may involve a transfer from USB and Bluetooth communications to Ethernet and Wi-Fi.

6.2.3 Mapping and Synthesis

Work on mapping and synthesis for the T-Stick will continue at least until the spring of 2008, since there is a committment to work with the performers and composers preparing for the Digital Orchestra final concert. To this end, further refinement of the collaborative mapping system described in Chapter 4 will also continue. The *Modal4* synthesizer constructed for the T-Stick has not yet been used in public performance, and it is inevitable that it's functionality will need to be augmented to meet the demands of composers and performers. As of the submission of this thesis, performers and composers will have been working with the modal synthesizer for approximately one month – work which will continue into the summer and the next year of the project. To some degree, composers have been left to decide which type of synthesis to use for their pieces, but it is hoped that a synthesizer custom-built for the abilities of the T-Stick will prove to offer much more flexibility and expressive potential than the previously-used options.

6.2.4 Vibrotactile Feedback

The addition of vibrotactile feedback to the T-Stick platform also holds interesting possibilities. In traditional acoustic instruments, vibrations are usually transmitted passively to the performer through the body of their instrument or through reeds or strings, and are intrinsically coupled to the excited state of the instrument and the sound being produced [55]. These vibrations are usually absent in DMIs, since the interface body is rarely connected directly to the sound-producing mechanism, but they can be simulated through active vibrotactile feedback [56].

Since vibration actuators can be heavy and power-hungry, vibrotactile feedback will first be incorporated into the "bass" version of the T-Stick. At 1.8 meters in length, the bass T-Stick is expected to be performed resting on the floor rather than supported by the performer, so heavy actuators and a dedicated power-cord should not adversely affect playing technique.

6.2.5 Testing

Finally, some objective evaluation of the success of the design and implementation would be extremely helpful not only in improving the T-Stick itself, but also to generally inform the DMI design process. Areas in which the T-Stick could help with research include studying the transferability of performer skill between members of the family, mapping experiments (to determine, for example, whether using a "physical" interaction metaphor is helpful for novice users exploring the instrument), and whether vibrotactile feedback is helpful in "feeling" the instrument's response. A DMI holds advantages over traditional instruments in this regard, since the mapping can be changed between trials to alter the responsiveness of the instrument or the relationship between the performer's gesture and audio or vibration feedback.

References

[1] S. Mann, "Natural interfaces for musical expression: Physiphones and a physics-based organology," in *Proceedings of the 2007 Conference on New Interfaces for Musical Expression (NIME-07)*, (New York, USA), pp. 118–123, 2007.

[2] E. R. Miranda and M. M. Wanderley, *New Digital Instruments: Control and Interaction Beyond the Keyboard*, vol. 21 of *The Computer Music and Digital Audio Series*. Middleton, Wisconsin, USA: A-R Editions, Inc., 2006.

[3] J. Ryan, "Some remarks on musical instrument design at STEIM," *Contemporary Music Review*, vol. 6, no. 1, pp. 3–17, 1991.

[4] International Conference on New Interfaces for Musical Expression. [Online]. Available: `http://www.nime.org/`. Accessed June 16, 2007.

[5] S. Oore, "Learning advanced skills on new instruments (or practising scales and arpeggios on your NIME)," in *Proceedings of the 2005 Conference on New Interfaces for Musical Expression (NIME-05)*, (Vancouver), pp. 60–65, 2005.

[6] E. M. v. Hornbostel and C. Sachs, "Classification of musical instruments," Translated from the original german by A. Baines and K. P. Wachsmann, *The Galpin Society Journal*, vol. 14, pp. 3–29, 1961.

[7] A. Eronen, "Comparison of features for musical instrument recognition," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, (New Paltz, New York), pp. 19–22, October 2001.

[8] A. Tanaka, "Musical performance practice on sensor-based instruments," in *Trends in Gestural Control of Music* (M. M. Wanderley and M. Battier, eds.), pp. 389–406, IRCAM – Centre Pompidou, 2000.

[9] I. Choi, "A component model of gestural primitive throughput," in *Proc. 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, (Montreal, Canada), pp. 201–204, 2003.

[10] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer Music Journal*, vol. 26, pp. 11–22, Fall 2002.

[11] J. Rasmussen, *Information Processing and Human-Machine Interaction: an Approach to Cognitive Engineering.* New York, NY, USA: Elsevier Science Inc., 1986.

[12] J. Malloch, D. Birnbaum, E. Sinyor, and M. M. Wanderley, "Towards a new conceptual framework for digital musical instruments," in *Proceedings of the International Conference on Digital Audio Effects (DAFx-06)*, (Montreal, Canada), pp. 49–52, Sept. 18–20 2006.

[13] D. Birnbaum, R. Fiebrink, J. Malloch, and M. M. Wanderley, "Towards a dimension space for musical devices," in *NIME '05: Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, (Vancouver, Canada), pp. 192–195, 2005.

[14] P. R. Cook, "Real-time performance controllers for synthesized singing," in *Proceedings of the 2005 Conference on New Interfaces for Musical Expression (NIME-05)*, (Vancouver, Canada), pp. 236–237, 2005.

[15] A. Mulder, S. S. Fels, and K. Mase, "Design of virtual 3D instruments for musical interaction," in *Proceedings Graphics Interface '99* (S. MacKenzie and J. Stewart, eds.), (Toronto, Canada), pp. 76–83, 1999.

[16] R. J. K. Jacob, L. E. Sibert, D. C. McFarlane, and M. P. Mullen-Jr, "Integrality and separability of input devices," *ACM Transactions on Computer-Human Interaction*, vol. 1, no. 1, pp. 3–26, 1994.

[17] P. Cook, "Principles for designing computer music controllers," in *Workshop in New Interfaces for Musical Expression*, (Seattle, USA), 2001.

[18] J. Ryan, "Effort and expression," in *Proceedings of the 1992 International Computer Music Conference*, (San Jose State University, USA), pp. 414–416, 1992.

[19] C. Cadoz, "Instrumental gesture and musical composition," in *Proceedings of the 1988 International Computer Music Conference*, (San Francisco, USA), pp. 1–12, ICMA, 1988.

[20] C. Cadoz and M. M. Wanderley, "Gesture-music," in *Trends in Gestural Control of Music* (M. M. Wanderley and M. Battier, eds.), pp. 71–93, Paris: IRCAM – Centre Pompidou, 2000.

[21] M. Waisvisz, J. Ryan, and N. Collins, "Twenty-five years of STEIM, an overture." report, 1993.

[22] J. Piringer, "List of electronic instruments." Online spreadsheet. Available: `http://joerg.piringer.net/`. Accessed June 16, 2007.

[23] R. Edgar, "The sweatstick." [Online]. Available: `http://rayedgar.com/sweatstick.html`. Accessed June 16, 2007.

[24] M. Cutler, G. Robair, and Bean, "The outer limits," *Electronic Musician*. August 2001.

[25] IDEO design, "Musical stick for interval research." [Online]. Available: `http://www.ideo.com/portfolio/re.asp?x=50032`. Accessed June 16, 2007.

[26] R. Tuttle, "The MUSICPOLE." [Online]. Available: `http://themusicpole.com/`. Accessed June 16, 2007.

[27] M. M. Wanderley, D. Birnbaum, J. Malloch, and E. Sinyor, "Sensorwiki.org: A collaborative resource for researchers and interface designers," in *Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME-06)*, (Paris, France), pp. 180–183, 2006.

[28] L. Bardos, S. Korinek, E. Lee, and J. Borchers, "Bangarama: Creating music with headbanging," in *Proceedings of the 2005 Conference on New Interfaces for Musical Expression (NIME-05)*, (Vancouver, Canada), pp. 180–183, 2005.

[29] E. Dykstra-Erickson and J. Arnowitz, "Michel Waisvisz: the man and the hands," *interactions*, vol. 12, no. 5, pp. 63–67, 2005.

[30] Q. R. Ghazala, "Circuit-bending and living instruments," *Experimental Musical Instruments*, vol. 9, pp. 21–23, 1993.

[31] L. K. Baxter, *Capacitive Sensors: Design and Applications*. IEEE Press Series on Electronics Technology, Piscataway, NJ, USA: IEEE Press, 1997.

[32] J. Fraden, *Handbook or Modern Sensors: Physics, Designs, and Applications*. New York, Berlin, Heidelberg: Springer, 3 ed., 2004.

[33] Q. R. Group, "Qprox(tm) qt160/qt161 6 key charge-transfer qtouch(tm) sensor ic." Datasheet, 2002.

[34] R. Koehly, D. Curtil, and M. M. Wanderley, "Paper FSRs and latex/fabric traction sensors: Methods for the development of home-made touch sensors," in *Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME-06)*, (Paris, France), pp. 230–233, 2006.

[35] M. T. Marshall, "Building a USB sensor interface," [Online]. Available: `http://www.sensorwiki.org`. Accessed June 16, 2007.

[36] "Bluetooth." [Online]. Available: `http://www.bluetooth.com/`. Accessed June 16, 2007.

[37] M. M. Wanderley, N. Schnell, and J. B. Rovan, "Escher - modeling and performing composed instruments in real-time," in *Proc. IEEE SMC'98*, pp. 1080–1084, 1998.

[38] A. Hunt, M. M. Wanderley, and M. Paradis, "The importance of parameter mapping in electronic instrument design," in *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02)*, (Dublin, Ireland), pp. 149–154, 2002.

[39] Cycling74, "Max/msp." [Online], Available: `http://www.cycling74.com/`. Accessed June 16, 2007.

[40] M. T. Marshall, N. Peters, A. R. Jensenius, J. Boissinot, M. M. Wanderley, and J. Braasch, "On the development of a system for gesture control of spatialization," in *Proceedings of the International Computer Music Conference*, (New Orleans), 2006.

[41] A. R. Jensenius, T. Kvifte, and R. I. Godoy, "Towards a gesture description interchange format," in *Proceedings of the 2006 Conference on New Interfaces for Musical Expression (NIME-06)*, (Paris, France), pp. 176–179, 2006.

[42] "The McGill Digital Orchestra." [Online]. Available: `http://www.music.mcgill.ca/musictech/DigitalOrchestra/`. Accessed June 16, 2007.

[43] J. Malloch, S. Sinclair, and M. M. Wanderley, "From controller to sound: Tools for collaborative development of digital musical instruments," to appear in *Proceedings of the International Computer Music Conference '07 (ICMC'07)*, (Copenhagen, Denmark), 2007.

[44] T. Place and T. Lossius, "Jamoma: A modular standard for structuring patches in max," in *Proceedings of the International Computer Music Conference*, (New Orleans, USA), pp. 143–146, 2006.

[45] "Integra: A composition and performance environment for sharing live music technologies," [Online]. Available: `http://www.integralive.org/`. Accessed June 16, 2007.

[46] M. Wright, A. Freed, and A. Momeni, "OpenSound Control: State of the art 2003," in *Proceedings of the 2003 Conference on New Interfaces for Musical Expression (NIME-03)*, (Montreal, Canada), 2003.

[47] A. Hunt, *Radical User Interfaces for Real-Time Musical Control*. PhD thesis, University of York, UK, 1999.

[48] J. Malloch, D. A. Stewart, F. Rocha, X. Pestova, A. Lindh, and R. Koehly, "MUGS695 final project documentation," project report, McGill University, April 2006.

[49] C. Dodge and T. A. Jerse, *Computer Music*. Schirmer, second ed., July 1997.

[50] T. Jehan, A. Freed, and R. Dudas, "Musical applications of new filter extensions to max/msp," in *Proceedings of the International Computer Music Conference*, (Beijing, China), pp. 504–507, ICMA, 1999.

[51] G. D'Arcangelo, "Creating a context for musical innovation: A NIME curriculum," in *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02)*, (Dublin, Ireland), pp. 102–105, 2002.

[52] "Schulich School of Music - Digital Composition Studios." [Online]. Available: `http://www.music.mcgill.ca/composition/Files/DCS/`. Accessed June 16, 2007.

[53] B. Bongers, "Physical interfaces in the electronic arts," in *Trends in Gestural Control of Music* (M. M. Wanderley and M. Battier, eds.), pp. 41–70, IRCAM – Centre Pompidou, 2000.

[54] J. A. Paradiso, K. Hsiao, J. Strickon, J. Lifton, and A. Adler, "Sensor systems for interactive surfaces," *IBM Systems Journal*, vol. 39, no. 3&4, pp. 892–914, 2000.

[55] C. Chafe, "Tactile audio feedback," in *Proceedings of the 1993 International Computer Music Conference*, (Waseda University, Japan), pp. 76–79, ICMA, 1993.

[56] B. Bongers, "Tactile display in electronic musical instruments," in *Colloquium Digest, IEE Colloquium Developments in Tactile Displays*, (London, UK), pp. 7/1–7/3, January 1997.