Chapter 6

Conclusions and Future Work

The research presented in this dissertation has focused on the potential role for computer-generated haptic feedback in interfaces for computer-based musical instruments. The work has sought to leverage the musician's existing sensitivity to the relationship between an instrument's "feel" and its sound. The objective of the experiments carried out was to discover whether adding haptic feedback to these instruments would improve their playability.

6.1 Designing Haptic Responses for Virtual Musical Instruments

6.1.1 The Virtual Theremin

Our starting point for these studies was to build a musical instrument with programmable haptic feedback that had no real-world equivalent. In this way, free from the possibility of interaction with playing technique for another instrument, we were able to explore different mappings between an instrument's auditory and haptic responses. In Experiment I (see Chapter 4) we measured the accuracy with which a player could play a melody under three feedback conditions: changes in force mapped directly to changes in pitch, changes in force dependent on velocity, and constant force feedback. Participants played a series of short melodies in each of six randomly presented feedback conditions. The results of this study indicated that performance was marginally better in those conditions where changes in force were directly correlated to changes in the parameter being controlled, in this case pitch. These results implied that haptic feedback that supported the objectives of the musical task could improve performance of that task.

The next question to address was whether these small gains in performance would be amplified or attenuated with practice. We hypothesized that, if players were given longer to "learn" the response of the instrument, the differences between correlated and uncorrelated haptic/auditory feedback would be greatly reduced or might even disappear altogether. Participants in Experiment II again played short melodies on the virtual Theremin, but were now assigned in advance to one of three haptic response conditions: correlated feedback (force changed as a function of position), uncorrelated haptic feedback (force changed as a function of velocity), and no feedback. Results showed that initially the differences between performance in the three conditions were similar to those in Experiment I. After a short period of practice, about 6 trials, players in both force-feedback conditions performed equally well, indicating that they had learned to compensate for the dynamics of the feedback condition to which they were assigned. However, players in the no-feedback condition showed only a small improvement in performance. Thus the presence of force feedback, not its specific character, was the factor that determined improvement in performance. We concluded that for these novice players, force feedback provided dynamic behavior that could be learned, augmented feedback through a second sensory channel informing players of the results of their actions. Taken together, these experiments indicate that providing some haptic response that is consistent with an instrument's auditory response enables the "feel" of an instrument to be more easily learned.

It is no accident that musicians consider the "feel" of an instrument to be as important as its sound. What they describe in an instrument's "feel" is how it responds to their actions, that is to say the consistency with which they can predict the relationship between actions they perform and the corresponding sounds that are produced. The implications of these studies are important because they suggest that, though programmable haptic feedback might tempt the instrument designer to experiment with instruments whose feel is continually evolving, players will find that these instruments are hard to learn and to control.

6.1.2 Simulations of Real Instruments: Experiments III and IV

Experiments with the virtual Theremin examined the potential benefits of adding haptic feedback to an instrument that, in the real world, provides no haptic cues to the player. However, mounting evidence from the literature suggests that, where a mechanical coupling between player and instrument exists, information in the form of both vibrations and reaction forces can provide the performer with cues about the instrument's state (see Chapter 5). For Experiments III and IV we constructed a virtual bowed string with haptic feedback that aimed to simulate normal and frictional forces present during bowing. The questions these experiments sought to answer were 1) Is the haptic feedback available via a mechanical coupling between player and instrument crucial to the control of the instrument? and 2) Is this haptic feedback part of a player's internal representation of the behavior of the instrument? In Experiment III, musicians with no experience of playing a bowed instrument imitated a prerecorded sample bow stroke as closely as they could. Half were given feedback about both normal and frictional forces while the second group felt only forces normal to the downward pressure of the bow. These bow strokes were then ranked by two independent observers. Results indicated that the presence or absence of friction in the haptic model had no effect on the "goodness" of bow strokes for novice players. We therefore concluded that feedback from friction between the bow and the string does not contribute to the novice player's ability to control a bow, at least in the early stages of learning.

In Experiment IV, experienced string players were recruited to play the virtual bowed string. Again, the "feel" of the model was manipulated in order to determine if the presence of friction would facilitate transfer of technique from a real to a virtual bowed instrument. Put another way, would the absence of friction affect in any way the player's ability to control the interaction between bow and string? Each player recorded two sequences of bow strokes, one with and the other without friction feedback in the haptic model. Again, their data were scored by two independent observers. The results of this study indicate that players did marginally worse when friction was present in the haptic model than when they could feel normal forces alone. From these results we concluded that our model was not realistic enough to promote the transfer of learned technique from the real to the virtual instrument, but that it was close enough to confuse players by interacting with their internal representation of the feel of the dynamics of bow-string interaction. These results imply that, unless simulations of instruments are very close in feel to their real counterparts, they are more likely to confuse players than to promote transfer of skill from the real environment. This having been said, most of the participants in this study indicated that they preferred the feel of the model when friction was present. Therefore, even if it was of no obvious benefit in performance, the presence of friction appears to have improved the quality of the simulation as perceived by these experienced players. In the impoverished world of appropriate controllers for computer-based bowed instruments, this would seem to be a step in the right direction.

6.2 Supporting Haptic Feedback in Interfaces for Computer-Based Musical Instruments

Though programmable haptic feedback was first incorporated into a computer-based musical instrument more than 20 years ago (Cadoz *et al.*, 1990), the technology of haptic display and the protocols that support haptic interaction are still relatively primitive. There is no manufacturer-independent protocol to support the connection of haptic display devices to Personal Computers, for example, and no standards for displaying haptic effects yet exist. How then might the technology used to create the instruments for this study be transferred from the lab into the hands of composers, performers, and instrument designers?

6.2.1 Hardware Considerations

Haptic feedback devices are most often configured as closed-loop devices, sensing the position of the operator's hand in the workspace and relaying forces based on this position back to the operator. The rate at which forces must be computed and updated is determined by our ability to sense the granularity in feedback and is accepted to be around 1kHz (Hasser and Massie, 1997). Given that devices need to be servoed at this rate, two configurations for incorporating haptic feedback into simulated environments currently exist. Either

1. control parameters derived from sensors in the haptic display device are fed at an appropriate sampling rate (usually 1KHz) to a central servo loop which generates force output based on these parameters, as in the case of the virtual Theremin, or

2. haptic feedback is computed on a separate processor, usually embedded in the device itself, which communicates with a control computer via an isochronous protocol, as is the case with the virtual bowed string.

Both approaches have advantages and disadvantages for music controllers. In the first case, the tight coupling between sound and touch provides the potential for a single physically based model of the instrument to drive both auditory and haptic feedback. Thus the frictional forces for a haptic rendering of bow-string interaction could be computed from the coefficient of friction generated as part of the audio model. Since movement is sampled at 1kHz, it is also possible to create an instrument that is responsive to tiny gestural nuances, giving the performer a sense of connection to the audio model that is lacking in existing control protocols such as MIDI. Currently, this approach is limited to very simple instruments, such as the Theremin, because the computational resources required to support both haptic and audio output from a single physical model are not readily available. Moreover, this approach requires haptic and auditory responses to be uniquely designed for each instantiation of the instrument, since they are highly dependent on each other.

For the virtual bowed string, therefore, we turned to the second approach and computed haptic and audio output on separate processors which communicated via MIDI. Here we were able to take advantage of MIDI's existing control protocol to communicate with an existing physical model of a bowed string (Serafin *et al.*, 1999). This modular design allowed us to experiment with different haptic responses for the instrument, but the position and velocity parameters, which were sampled locally at 1kHz, had to be subsampled to be transmitted via MIDI so that the violin model was only updated every 200msec. This process inevitably introduced a small amount of latency, which experienced players could easily detect.

The most advantageous approach is therefore a hybrid approach in which haptic and auditory models can communicate at a rate of 1kHz, either by inter-process communication on a single machine or by high-speed hardware communication. This approach leverages both a high-bandwidth connection to capture nuances of gesture and a modular design to allow for redesign or substitution of either part of the model. With this design, haptic controllers can be thought of as gestural controllers, generating sampled signals that can either operate on synthesis parameters directly or can be analyzed and parsed into events. The requirements for communicating with haptic devices in the context of gestural control of music are stringent. Not only must a hardware protocol support two-way communication at 1kHz, but it must also fulfil the requirements for any gestural controller intended for live musical performance. As enumerated by Roads (1996), these include support for isochronous communication, electrical isolation, transmission over distances greater than 50 meters, and gesture sampling rates of over 100 kHz. (For an excellent review of current high-bandwidth protocols in relation to gestural control of music see Fried and Wessel, 1998.) However, even meeting the most stringent hardware requirements in a low-level hardware protocol is of little use without a control protocol that can support the communication of complex gestures and the relaying of the response of a mechanical system to these gestures (see Machover, 1992, for a discussion of hardware and software support for complex gestural controllers). Taking MIDI as a starting point, we next consider the implications for a control protocol of supporting haptic interaction in music controllers.

6.2.2 Implications for the Design of a Communication Protocol

The goal of any protocol for communicating between a music controller and a synthesis module is to support the translation of musical nuance, expressed as movement, into parameters for controlling sound. Thus the control protocol becomes an interpreter, mediating between the performer's intent and the synthesis module's realization. The question of how to design such a protocol has vexed computer music research for almost 30 years, but has become more prevalent in an age where real-time gestural control is the de facto standard for performance systems.

Currently, the protocol for interconnection of digital musical instruments that is supported by industry is MIDI (Musical Instrument Digital Interface.) Though far ahead of its time in even implementing programmable real-time control, MIDI was early constrained by pressure from industry to adopt the piano keyboard as the default control device. Thus the protocol developed around a paradigm of ballistic control with few control parameters affording the opportunity to close the loop between performer and instrument. Along with the constraint of the keyboard, MIDI also inherited the metaphor of the recording studio. Sequences of control parameters, transmitted on "channels," could be recorded as "songs" with related instrumental "parts" grouped onto "tracks," much as percussion and string sections might be laid down on separate tracks in a conventional multi-track recording studio. The tug-of-war between demands for a flexible real-time communication protocol for expressive performance on the one hand and demands for an all-purpose protocol digital mixing paradigm on the other have produced a complex protocol that cannot fully support either of its target user groups.

In terms of communicating musical nuance, MIDI breaks down at many levels. Purely in terms of hardware, it cannot support isochronous high-speed two-way communication between controller and instrument. But more importantly, even if hardware constraints were removed, the command structure of MIDI pays little attention to the hierarchical nature of musical performance. As noted earlier (see Chapter 3), expressive musical ideas, and their attendant sequences of movements, are organized by the human motor system into hierarchies. Low-level events, such as the execution of individual notes, are presumed to be encapsulated in motor programs which can be triggered in sequences or patterns that are also learned. But these higher-level patterns can be shaped and reshaped in real time by meta-level movement control. Thus rubato, ritardando, and accelerando gestures can be superimposed at will on music that has already been learned. For a protocol to successfully support gestural control, therefore, there must exist an architecture to support this hierarchy, taking into account the "connections" between various control layers. Because the command structure of MIDI ignores this hierarchy of control in music, it is not capable of translating performance gestures into musical nuance (see Appendix C for a classification of MIDI commands by functional level of control).

If a protocol is to simulate the mechanical coupling between player and instrument, transmitting movement and relaying reaction forces, then support for meta-level gestural control becomes even more important. A sequence of notes played with one bow stroke, or governed by one arm movement in the case of keyboard technique, is bound together by a fluid movement that requires a coherent response from the instrument. Without support for groupings defined by movement in this way, it is unlikely that a close coupling between action and response for an instrument with haptic feedback can be maintained. If the response of the instrument is unpredictable or unstable, then the illusion of a direct mechanical coupling will break down altogether and the instrument will be unplayable.

6.2.3 Development Environment for Haptic Feedback

If we assume that both a hardware protocol and a supporting command structure for gestural control exist, there still remains the challenge of providing tools for the design of computer-based musical instruments with programmable haptic feedback.

One constraint is that haptic display devices lack the common functionality of the computer screen for graphics or the loudspeaker for sound. Some devices, such as actuated exoskeletons or gloves, aim to convey virtual haptic images by intervening at the boundary between the hand and its environment, while others convey a haptic impression of the environment as it might be experienced through a tool such as a gripper or stylus. Tools for designing haptic interactions for computer-based musical instruments must take into account the varying affordances and constraints of haptic display devices that are currently available, and must be flexible enough to support the development of new devices.

Another limitation is that we only partly understand the role played by haptic feedback in music performance. Much experimental work yet remains before we can fully utilize the haptic channel to control complex synthesis modules. For such work to be possible, it is necessary to understand the relationship between actions, in the form of expressive gestures, and haptic responses; this understanding would make it possible to generate responses appropriate to the gestures. A development environment must therefore allow for substitution of haptic display hardware, communication protocols, and haptic rendering modules with relative ease.

Finally, a development environment should provide instrument builders with access to algorithms for rendering haptic effects, making available primitives such as springs, dampers, and friction effects, which can be used as building blocks for haptic interaction design. Though individual manufacturers do make such development environments available for their own devices, there is as yet no unified development tool for haptic interaction design.

The need for platform- and hardware-independent development environments is one that is familiar to the field of computer music. For almost forty years, composers and performers have been creating environments in which synthesis algorithms and techniques can be shared that are independent of both the host platform and the sound synthesis hardware. For a small, widely distributed user community, such environments represent knowledge banks and are key to the survival in the public domain of valuable synthesis algorithms. In this work, we have used the "synthesis tool kit" (Cook and Scavone, 1998). In this environment, hardware devices and drivers, software simulation, and communication protocol are all separate modules, allowing for rapid prototyping on different platforms, with different communication and control hardware. The synthesis tool kit (STK) is mostly open-source, though much of the low-level protocol and hardware support, including some manufacturer-specific drivers, are distributed in precompiled libraries. Because the environment is completely modular but can still protect the interests of hardware and software manufacturers, it would seem to be a useful model on which to base a development environment for haptic interaction design. As we move toward incorporating haptic interaction into multi-modal interfaces, such an approach would provide an environment in which hardware and software developers, interaction designers, and end users could coexist.

6.3 Summary

The interaction between musician and instrument in the context of musical performance is predicated on the player's understanding of the highly complex dynamics of the instrument. Players perform actions defined by expressive musical goals, with the expectation of a performance that reflects these goals in nuances of timing, timbre, and dynamic contour. The principal source of feedback informing musicians of the results of his or her actions is the instrument's sound. However, where it exists, a mechanical coupling between player and instrument can also convey information about the instrument's state that can be felt. Sensory feedback appears to play its most important role in the early stages of learning when musicians are building an internal representation of the link between his or her actions and the instrument's response. But evidence presented earlier in this work suggests that sensory feedback is important for experienced musicians, too, alerting them to unpredicted changes in the instrument's state.

Providing a second channel of sensory feedback in interfaces for computer-based musical instruments would therefore seem to be of benefit to novice players and experienced musicians alike. The results of the studies presented in this work indicate that, to be of use to the musician, the haptic response of a computer-based instrument must be predictable and stable. In those cases where the simulation is of an existing acoustic instrument, simulations must be of high fidelity if they are to promote transfer of skill from the real to the virtual environment.

Based on experience gained in developing the instruments for this study, we suggest that there are three goals that need to be met if computer-based musical instruments with programmable haptic feedback are to become a reality. Firstly, the communication between a controller with haptic feedback and the synthesis module it controls must be supported by hardware that allows for two-way high-bandwidth isochronous communication. Secondly, the protocol that translates movements sensed at the controller into complex musical gestures must support a hierarchy of commands that can operate on music at many levels, from the articulation of individual notes to the shaping of musical phrases and subphrases. Lastly, an environment must be created in which performers, composers, and instrument designers can explore the possibilities afforded by this new modality for musician/instrument interaction design.

6.4 Future work

The work presented here has begun to explore the role for haptic feedback in the environment of musical performance. Experiments have endeavored to discover what role haptic feedback plays in the process by which the musician builds a mental representation of the dynamics of an instrument. As such, these studies have only begun to reveal the processes by which we learn to play new musical instruments, whether learning new techniques or transferring techniques we have previously learned. One future research goal is to develop experimental techniques that can further uncover the processes by which we build internal representations of the behavior of tools we use in the real world. This will enable appropriate simulation of the haptic component of these interactions in virtual environments. Developing the virtual instruments used in these studies was a long and laborious process. Finding a configuration that could satisfy the constraints imposed by supporting haptic and audio simulations and the protocol by which they communicated required much experimentation. A second research goal therefore is the development of an environment for designing computer-based musical instruments that integrates support for audio and haptic simulation in real time, an environment in which composers, performers, and musical instrument designers can explore the possibilities afforded by this newly available sensory modality for computer-based instrument design.