A hybrid keyboard-guitar interface using capacitive touch sensing and physical modeling

Heinrichs, C; McPherson, A; Sound and Music Computing

For additional information about this publication click this link.
http://qmro.qmul.ac.uk/jspui/handle/123456789/4241

A Hybrid Keyboard-Guitar Interface using Capacitive Touch Sensing and Physical Modeling

Christian Heinrichs
Queen Mary University of London
`christian.heinrichs@eecs.qmul.ac.uk`

Andrew McPherson
Queen Mary University of London
`andrew.mcpherson@eecs.qmul.ac.uk`

ABSTRACT

Physical modeling allows realistic guitar synthesis incorporating many expressive dimensions commonly employed by guitarists, including pluck strength and location, plectrum type, hand damping and string bending. Often, when a physical model is used in performance, most control dimensions go unused when the interface fails to provide a way to intuitively control them. Techniques as foundational as strumming lack a natural analog on the MIDI keyboard, and few digital controllers provide the independent control of pitch, volume and timbre that even novice guitarists achieve. This paper presents a hybrid interface based on a touch-sensing keyboard which gives detailed expressive control over a physically-modeled guitar. Most dimensions of guitar technique are controllable polyphonically, some of them continuously within each note. Mappings are evaluated in a user study of keyboardists and guitarists, and the results demonstrate its playability by performers of both instruments.

1. INTRODUCTION

Physical modeling allows detailed simulation of familiar acoustic instruments and the creation of novel instruments, even physically impossible ones. But in performance, a good model is only half the challenge: the control interface and the mapping from gesture to sound strongly influence the resulting interaction. Gelineck and Serafin [1] propose a set of directives for controlling physical models with the goal of encouraging creativity and exploration. Among these are a balance between the model's sonic diversity and plausibility, the control of physical models with physical gestures, and experimentation with the interplay between instantaneous and continuous gestures. They describe a modular system in which, among other things, gestures from one instrumental technique can be coupled to sound sets associated with another instrument, producing new hybrid musical systems.

In this paper we describe the development of one such cross-instrumental interface. A physical model of a guitar is proposed whose control parameters intuitively relate to the actions of the left and right hands in guitar playing. We are interested in encouraging creative exploration of these parameters, and also in transferring the experience of guitar playing to musicians with little or no guitar training. Traditional keyboard interfaces, even when coupled to sophisticated synthesizers, largely fail to capture this experience: techniques as foundational as strumming have no keyboard analog, the distinction between pitch selection and note activation does not exist, and expressive dimensions including pluck location and string bending are difficult to achieve. On the other hand, GUI interfaces lack the physicality of instrumental playing.

To provide both physicality and multidimensionality, we control the physical model using a touch-sensitive musical keyboard we recently developed [2] which measures the location and contact area of fingers on the key surfaces. Mapping between touch data and physical model parameters thus becomes a primary focus for study, with the goal of providing intuitive control to guitarists and non-guitarists alike. Our goal is not to replicate the guitar; rather, we seek to explore the creative possibilities of a hybrid instrumental technique while providing an instrument non-guitarists can use to produce common guitar techniques.

In the remainder of this paper, we examine related work and present our physical model, with a focus on the relevance of the control parameters to instrumental technique. We then discuss how these parameters are controlled by physical gestures extending traditional keyboard technique. Our results are evaluated in a user study of 10 keyboard and guitar players.

1.1 Related Interfaces

Our work connects to a long tradition of extended keyboard interfaces. The keys of the early 20th-century Ondioline [3] could be moved horizontally to create vibrato effects. Moog and Rhea created a keyboard which sensed the position of fingers on the key surface, much like our present design [4]. Other recent extended or abstracted keyboards includes the Continuum [5], the Seaboard [6], the Hyperkeys [1] and the Endeavour Evo [2] .

Our goal of simple, expressive guitar-like sounds also has a venerable history. The autoharp and its electronic successors including the Suzuki Omnichord are expressly designed to simplify chordal playing. More broadly, many graphical and physical interfaces, including the Ghost controller [7] and Apple's GarageBand for iPad, are designed around the principle of separating chord selection and string

---

[1] `http://www.hyperkeys.com/`
[2] `http://www.endeavour.de/`

activation. The Kalichord [8] achieves a wide tonal range by driving physical string models with signals from piezo tines plucked by the performer. The performer's other hand controls a series of buttons selecting the string tunings.

In comparison to previous work, this paper seeks a more explicit hybrid of control gestures from keyboard and guitar. It also offers many degrees of expressive control in the context of a detailed physical model whose parameters are intuitive for musicians to understand.

2. PHYSICALLY-MODELED GUITAR STRING

2.1 Core Synthesis

The aim was to create a realistic and computationally efficient model of a guitar that is controllable by parameters that have an intuitive relationship to the player. The model developed for this study is based on Karjalainen, Välimäki and Tolonen's extensions of the Karplus-Strong algorithms [9], shown in Figure 1.
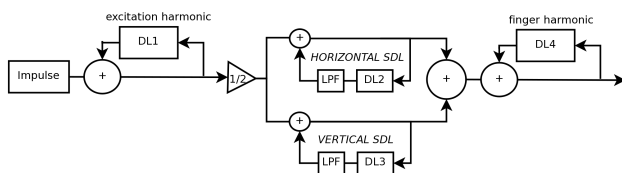
Figure 1. Signal chain of guitar model.

2.1.1 Single Delay-Loop model

The propagation of energy along a string in both directions can be simplified to a single delay-loop (SDL) model. This consists of a simple delay line with a low pass filter integrated into the loop circuit, of which the Karplus-Strong algorithm can be seen to be a special case. Two SDLs lie at the center of the model corresponding to the vertical and horizontal motion along the string.

2.1.2 Harmonics

Two further delay lines (*DL1* and *DL4*) are integrated into the signal chain to simulate excitation position and left-hand harmonics.

The frequency is controlled by calculating the exact position on the string, which, in the case of the finger harmonic, relates to the corresponding note that would sound if the finger were held down firmly. The held note can be obtained by taking the inverse of the string length (1/L) and multiplying this by the fundamental frequency. The harmonic is obtained by calculating the ratio from the string length between the finger and the nut to the entire string length and multiplying the inverse by the frequency.

2.1.3 Excitation

To reduce computation, wavetables from recorded impulse responses are used to excite the signal chain. Six impulse responses were recorded at the bridge, the body and the fingerboard of an acoustic guitar. Three of these were done by striking a muted string with a plectrum and three using a fingertip. While this creates a realistic impression of

a guitar body it is not a physical simulation. This means that in order to adjust any sounds resulting from the guitar body (e.g. resonances at particular frequencies) new sets of impulse responses need to be recorded. In our implementation we convolved these wavetables with freely available room impulse responses as an efficient way of achieving reverberation.

2.1.4 Coupling

On real instruments, vibrations from one string will produce sympathetic resonances in the others. The model uses a two-dimensional waveguide, and we use the output from the vertical SDL of each string to resonate the horizontal SDL of each other string, thereby avoiding feedback problems. This leaves us with six inputs and six outputs for each string. These can be routed to each other by a 6x6 matrix. The volume of each output scales with the consonance of the frequency ratio between strings. The frequency ratio between driving and observed strings is calculated; the closer it is to a multiple of 0.5, the higher the amount of consonance. All couplings can be globally scaled as an adjustable parameter, with small coupling values producing the best results.

This is an efficient albeit simplified way of recreating string coupling effects without relying on precise but computationally expensive techniques. For more detailed implementations see [10] and [11].

2.1.5 String Variance

Our model makes further use of vertical and horizontal waveguides by adding a slight random variance to the cut-off frequency, feedback level and pitch of the respective SDLs. The resulting sound qualities are a less predictable, more exponential decay envelope and a subtle beating effect caused by the two frequencies that are slightly out of tune. The amount of variance is an adjustable parameter in our model.

2.2 High-Level Parameters and Control

2.2.1 Changing Notes

To simulate changing notes on a string without introducing artifacts in the output, feedback values of the delay-lines are ramped down and back up again as the frequency is changed. Simultaneously, the transition between frequencies must occur with sufficient amount of low-pass filtering to avoid both abrupt artifacts and undesired portamento. Suitable parameters were chosen by trial and error and illustrated in Figure 2.

2.2.2 Pitch Bend

The resonant frequency of the delay lines can be changed to bend a note's pitch. Low-pass filtering of the frequency parameter is used during a bend to avoid zipper noise and other interference.

2.2.3 Finger Pressure

To extend on this particular focus was placed on the simulation of finger pressure while selecting notes on each
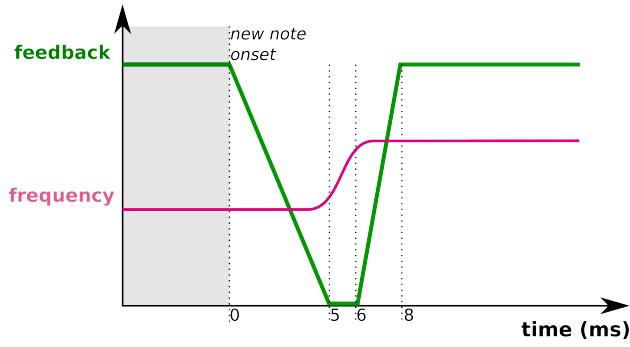
Figure 2. Feedback envelope.

string. Pressure parameters between 0 and 100 percent were mapped to feedback levels of the main horizontal and vertical delay-loops of each string as well as to the comb-filter for generating finger harmonics. A pressure value of 100 percent corresponds to having a finger firmly pressed down behind a fret, resulting in a long decay time. As this value decreases to 50 percent the the sound the decay time drops significantly to the extent that the pitch is almost unidentifiable. However, at a further slight decrease in pressure a harmonic is activated corresponding to the fret position. Again, the overall string decay time is reduced as the pressure value approaches zero.
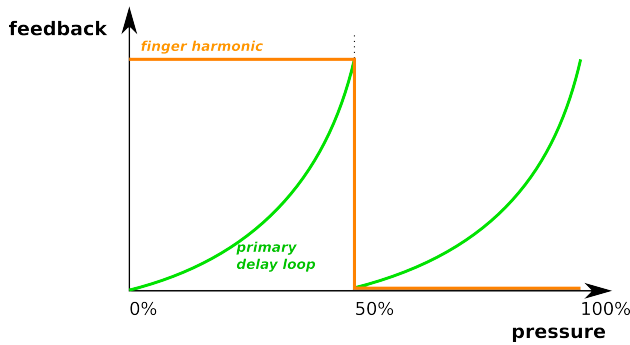
Figure 3. Pressure function.

2.2.4 Decay

The maximum feedback level for all primary delay loops, which controls string decay, is adjustable within a range of 0.99 to 0.99999. This is mapped to a more intuitive 'decay' parameter ranging from 0 (fastest decay) to 1.

2.2.5 String Damping and Palm Muting

To simulate the effects of string tension and thickness when stopping notes on a string, the cut-off frequency of the low-pass filter in each SDL is set to a multiple of the frequency it is tuned to. By adjusting this constant by which tuned frequency is multiplied, all strings can be dampened uniformly. A constant greater than 100 produces a bright sound with a long decay, whereas values as low as 4 produce sounds that resemble the effect of palm muting on a guitar.

2.2.6 Further Exploration

Recent studies have investigated more detailed simulation of the interaction between player and guitar. These include the employment of scattering junctions to simulate the neck-side hand interaction with the fingerboard [12] and the implementation of string friction while sliding between fret positions [13]. These advanced features have not been implemented in this study, as the aim was to create a realistic guitar model with a sufficiently low level of CPU usage and enough modularity to explore mapping strategies for real-time control.

2.3 Summary of Parameters

By using the SDL model we can get closer to the real sound of a plucked string by simply adding components, rather than deriving precise motions and parameters of waveguides from complex real-life examples. This modularity makes it particularly valuable for this study as components can easily be added or modified to suit the interface.

The guitar model that we used in this study responds to the following parameters:

- String tuning
- Fret selection and note changing
- Pitch bending
- Palm muting
- Left-hand pressure
- Excitation position and type
- String variance
- Coupling intensity
- Decay
- Reverberation

3. TOUCH SENSITIVE KEYBOARD CONTROL

The parameters of the physical model are intended to have intuitive meaning for musicians, and we seek to extend this intuition to the performance interface. Using a multi-touch-sensitive keyboard, we developed mappings which combine familiar modes of keyboard playing with novel physical gestures related to the experience of guitarists.

3.1 Hardware

In recent work [2], we developed a keyboard incorporating capacitive touch sensing on the surface of every key (Figure 4). The keyboard measures the location and contact area of each finger at a rate of up to 200 samples per second. The touch data is transmitted to a computer using USB, where host software generates OSC and MIDI messages. Making intuitive use of multiple continuous control dimensions is a challenge for any interface; in this paper, we explore how mapping the touch-sensor data can create a playable, expressive virtual string instrument.
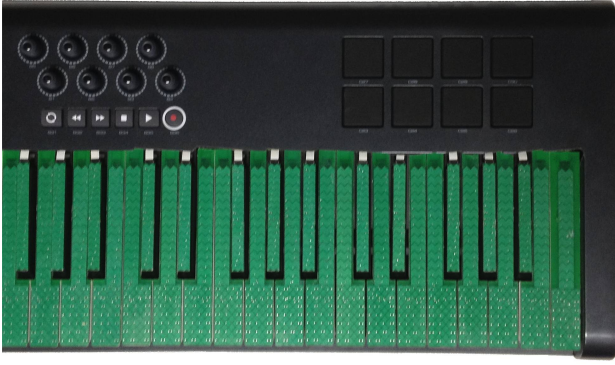
Figure 4. Capacitive touch sensors on a MIDI keyboard.

3.2 Design of a Hybrid Interface

Starting from the physical model in Section 2, our objective was to create an instrument that combines the playing techniques and expressive affordances of the keyboard and the guitar. Our design reflects the following goals:

- Continuous control over each note
- Intuitive to play for both keyboardists and guitarists
- Emulation of common guitar practices like strumming and string-bending
- Easily learned by beginners but expressive in the hands of experts

The target user community includes both pianists and guitarists, particularly musicians who are familiar with both instruments but not necessarily expert at either.

3.3 Parameter Mapping

The core of the instrument design is the mapping between actions on the keyboard and parameters of the physical model. We explored two approaches to parameter mapping, one modeled on traditional keyboard technique and one modeled on the distinction between left and right hands in guitar playing. We intend both to be used by the same performer, depending on the musical context.

3.4 Note Mode: Translating Constraints

The first mapping preserves the "one key, one note" principle of the keyboard. Following keyboard tradition, each key triggers a string pluck of the corresponding pitch with key velocity mapped to pluck strength, but unlike typical synthesizers, we focused on translating actions on the keyboard to the physical state of a virtual instrument, engaging with the mechanical constraints of the guitar in an effort to produce more realistic behavior.

The six strings of a guitar define both its sonic character and a set of constraints. On six strings, only certain combinations of notes can be played at once, and the choice of string affects the timbre of the note. Melodic passages sound different depending on whether they are fingered on the same string or played across multiple strings. By translating certain of these constraints to the keyboard, it is possible to obtain a more characteristic guitar response. Our

approach to mapping balances these goals against the need to support the intuition of keyboardists.

3.4.1 Note Allocation

Our physical model maintains a fixed number of strings. When a key is pressed, the note is allocated to the highest-tuned string capable of playing it (since fingering a fret can only increase the pitch of a string). Simulating specific fret positions on a fixed collection of strings can increase the realism of the model, especially when vibration coupling between strings is involved (Section 2.1.4).

Fixing the number of strings also preserves certain constraints of the guitar, including the lowest playable note and the combinations of notes that can sound simultaneously. In this project, we investigated whether translating the guitar's constraints to the keyboard would improve the realism of the playing experience or end up being a distraction. After an initial six-string implementation, we found that many common keyboard voicings were impossible to play, so we settled for an intermediate approach using three identical copies of each string (18 total), which preserves the benefits of cross-string coupling while allowing nearly any passage to be played. User reactions to this configuration are further discussed in Section 4.

3.4.2 String Bending

Continuous finger position sensing allows the player to bend the pitch of each note. Both upward and downward bends are found in guitar technique, the former by pushing the string sideways with the left hand, the latter using a whammy bar to reduce string tension. In our mapping, the initial key press always sounds at the expected pitch, regardless of finger position; lengthwise finger motion after note onset bends the pitch up or down (Figure 5).
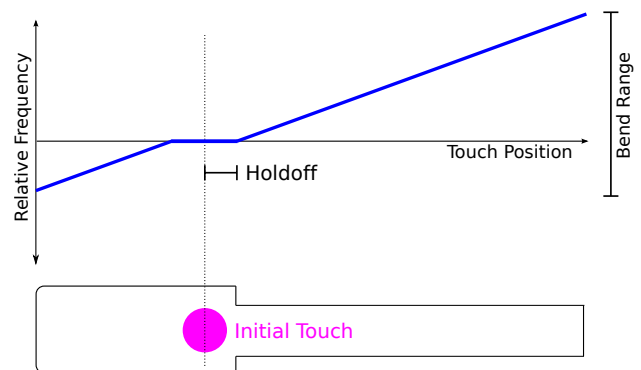
Figure 5. The performer bends the pitch of a note by moving the finger after note onset. Holdoff and range parameters are adjustable.

The total *range* of pitch bend per key is adjustable. Because the fingers move slightly on the keys during normal playing, we also implemented a *holdoff* parameter, expressed as a fraction of the total key length. Pitch changes only begin when the finger moves outside the holdoff zone. In user studies (Section 4), we investigated performers' preferences for both range and holdoff values.

3.4.3 Pluck Type

Globally, the instrument can be configured to use a plectrum or a fingertip model to activate the strings. As outlined in Section 2.2.4 three impulse responses were recorded for each excitation type: at the bridge, the sound hole and near the fretboard. The touch position is mapped to the excitation position of the guitar, which affects the corresponding harmonic and mix of impulse responses.

Guitarists also use palm muting as an expressive parameter, producing notes with rapid decay. We emulate this capability using the contact area between finger and key to control the damping factor (Section 2.2.5). Small contact area resulting from playing on the fingertip or fingernail produces a more muted tone compared to playing on the pads of the fingers. The mapping is designed so that normal playing position produces the common unmuted guitar sound, requiring a deliberate expressive decision to achieve the muted sound.

3.5 Strum Mode: Repurposing the Keyboard

The biggest gap between the traditional keyboard and the guitar relates to the guitarist's right hand: there is no physical analog to strumming on the keyboard. Emulating the back-and-forth strumming patterns of guitarists on a keyboard is challenging and generally unsatisfying for beginner and expert pianists alike. On the other hand, beginning guitarists may struggle with the left hand fingering needed to accurately render chords.

Strum Mode uses the touch sensors to bring strumming capability to the keyboard (Figure 6). Six white keys (C to A) in the top octave represent the strings of the guitar. The string is plucked when the performer touches the key surface. By not requiring full key presses, the performer can strum the strings by running the hand back and forth along the top of the keys, and because the keys are mechanically distinct, the arrangement also provides tactile feedback usable both for strumming and for fingerpicking (selecting individual strings).
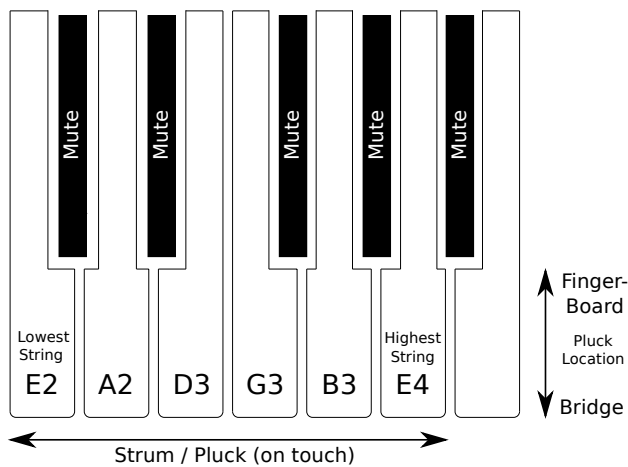
Figure 6. Mappings of guitarists' right-hand actions in Strum Mode, in which 6 keys are used to control 6 virtual strings. Chords are selected by the keyboardist's left hand.

3.5.1 Chord Selection

Strum Mode is designed to simplify chordal playing. The performer selects a chord by holding down the corresponding note in the left hand while strumming with the right. Single notes by default select major chords; minor chords and seventh chords are played by holding down multiple notes which outline the chord (see Table 1 for examples). Chords are voiced as they would be fingered in standard guitar tuning. Since not all six strings are used in every chord, certain strings are muted for selected chords.

When the left hand is released, the strings are damped as a guitarist would by holding the left hand loosely over the fingerboard. Strumming without a chord selected will produce a pitchless attack usable for common rhythmic effects employed by guitarists. Strings can also be damped by touching the top octave black keys, analagously to a guitarist muting the strings with the right hand.

Keys	E2	A2	D3	G3	B3	E4
C (E) (G)	X(0)	3	2	0	1	0
C E♭ (G)	X(3)	3	5	5	4	3
D (F♯) (A)	X(2)	X(0)	0	2	3	2
D F (A)	X(1)	X(0)	0	2	3	1
E (G♯) (B)	0	2	2	1	0	0
E G (B)	0	2	2	0	0	0
E (G♯) (B) D	0	2	0	1	0	0
E G (B) D	0	2	0	0	0	0

Table 1. Examples of mapping keys pressed to fret numbers on six virtual strings in standard guitar tuning. Keys in parentheses are optional. 0 indicates an open string; an X indicates a muted note, not normally strummed.

3.5.2 String Bending

Like the previous mapping, strings can be bent up or down in pitch. Bends are controlled from the right hand: when a key corresponding to a string is pressed down, deviation in front-back position will alter the pitch of the string. The requirement to press the key rather than just touch it ensures that the bend is a deliberate action.

3.5.3 Pluck Location

Guitarists use pluck location as an expressive dimension: plucking or strumming near the bridge will produce a brighter, thinner tone than corresponding actions near the fingerboard. For the keys controlling the six strings, we map the location of the initial touch to pluck position, with the frontmost edge of the key corresponding to the bridge. This allows more tonal variety than can be achieved on conventional guitar synthesizers, but the timbre remains similar enough in all positions that the instrument can be played without precisely controlling this parameter.

4. USER STUDY

Our mappings were designed to produce a realistic, playable instrument capable of techniques used on both keyboard and guitar. To evaluate our results, we tested the interface

with 10 musicians. Four users were keyboardists with less than 1 year experience with guitar; two users were guitarists with little or no keyboard experience; four users were experienced on both instruments. Overall, participants with keyboard training reported a mean of 14.4 years experience and a self-rated expertise of 5.6/10 (where 1 is complete beginner, 10 is top-level professional). Participants with guitar experience reported a mean of 13.9 years experience and an expertise of 6.4/10.

In individual sessions of roughly 35 minutes, performers were given a chance to explore the instrument, evaluate its usability and suggest alternative mapping strategies.

4.1 Reactions to Note Mode

4.1.1 Gesture and Intuition

Each participant was initially told only basic information about the instrument: that it was a physically-modeled guitar controlled by a touch-sensitive keyboard. Participants were asked to explore the instrument for several minutes and their reactions observed. 7 of 10 users discovered note-bending within the first 30 seconds of playing without being told of its existence. [3] In a typical example of the exploration process, one user uninentionally caused a note to bend, expressed surprise at the result, then began to deliberately explore the technique. By contrast, only 1 of 10 users discovered the ability to control string muting (Section 3.4.3) as an expressive parameter.

Each participant was asked to demonstrate the gesture they would most naturally choose to create vibrato on a note. 8 out of 10 users responded with the same gesture: rocking the hand side-to-side with the finger held in place on the key. One user chose a similar motion but with the hand turned sideways, so the rocking motion went back and forth along the key. The final user, a pianist, suggested the use of a pedal.

Participants were then asked to choose a gesture to represent bending a note substantially upward or downward in pitch. Here, 8 of 10 users chose front-to-back finger position (as the instrument was already configured) rather than side-to-side motion. Three of these users explained that sideways motion would be more intuitive but that the geometry of the keyboard favored vertical motions. One user argued for sideways motion regardless of this constraint, and the final user (the same pianist above) suggested a pedal.

Users were generally satisfied with the strategy of using finger position relative to note onset to control pitch. Several users felt that it would take time to become comfortable controlling the pitch bends, but all felt that it could be learned. One user suggested that only a certain range of the key be made available to pitch bends to reduce the possibility of accidental triggering. Reactions were mixed to the string muting mapping. 5 of 10 users found the mapping musically useful, but even within that group, most found the required finger curvature awkward. Two users suggested that a binary (muted or unmuted) mapping would be easier to control than a continuous variable.

---

[3] Some users had previously seen the touch-sensitive keyboard, though none had seen this particular mapping.

4.1.2 Fine-Tuning Parameters

We used the user study to determine the optimal values of several parameters. The string decay rate is adjustable in the physical model, and we asked each performer to rate which setting produced the most realistic guitar sound. A decay parameter of 0.7 was the consistent favorite, though some participants observed that the value would be different for acoustic and electric guitars.

Pitch bending was seen by most participants as one of the most valuable features. We asked each participant to rate the best value of the holdoff and range parameters (Section 3.4.2). Choices for holdoff were 0 (no holdoff), 0.1, 0.2 and 0.3 (expressed as fractions of the total key length). 8 of 10 participants chose 0.1 as the best balance between sensitivity and avoidance of accidental pitch bends; the remaining two chose 0.2. There was universal agreement that 0 (no holdoff) was unplayable, an important result for designers of continuous control surfaces as it suggests some tolerance of unintentional motion is necessary.

Participants were given a choice of the total pitch bend range of 2, 5 or 12 semitones, describing the range from lowest to highest achievable pitch. Most participants preferred a range of 5 semitones, with three participants preferring the smallest range. Most felt that a smaller range gave them more accurate control, but that a range of 2 semitones made it difficult to achieve common effects like bending notes from the lowered 7th scale degree up to the octave. Several users noted that a smaller bend range was more realistic to guitar playing, but that they found a larger range preferable in a novel instrument.

4.2 Reactions to Strum Mode

After participants explored the keyboard in note mode, it was changed to strum mode (Section 3.5). The basic premise of strumming the strings was explained, at which point the participant was encouraged to explore the instrument without further guidance.

4.2.1 String Activation

Nearly every user understood both strumming and chord selection concepts right away. Explorations included one- and two-directional strumming and tapping individual keys in a fingerpicking pattern. We received many comments during this process, most of them positive. One user commented "I'm not a guitar player but I can feel the sense of playing the guitar." The users who reacted more negatively focused on the feel of the keyboard. Two users found it uncomfortable or painful to run the hand along the edges of the keys, a mechanical issue we will address in a future version of the interface through better sanding and coating. Another user focused on technique: "As a guitar player, your right hand has to be a percussion instrument, and this gesture [swiping across the keys] doesn't feel like percussion." The same user later found fingerpicking more usable: "That's actually quite doable, that surprised me.... That's actually more like what playing is compared to strumming."

We did not find any pattern in whether keyboardists or guitarists reacted more positively to this mapping. Key-

boardists tended to have a greater expectation that the keys would need to be pushed down for an action to happen, especially for fingerpicking patterns, even though the initial touch triggers the pluck. In general, after pitch-bending, strumming was seen as the interface's most useful feature.

4.2.2 Chord Selection

Single key presses in the left hand map to major chords, which is where every user began. During the exploration process, three users discovered minor chords without being prompted. Users were later asked how they would expect to generate a minor or seventh chord; three more users immediately chose to finger the chord in the left hand, as our mapping operates. When the mapping was explained, all users found it to be intuitive and playable. Likewise, most users reacted positively to the unpitched plucking sound when no keys were held in the left hand, with several using it for rhythmic effects.

4.3 Further Comments

We asked each user how they would switch between mapping modes in performance. Most felt that a button on the top of the keyboard would be sufficient, though some suggested that the two modes could be combined to operate simultaneously, which we will explore in future work.

For performers with experience on both keyboard and guitar, we asked which technique the instrument related more closely to. Answers were surprisingly varied, with one user indicating keyboard technique, one indicating guitar technique, and the other two saying it drew on both. Though the sample size is too small to be statistically significant, we take this as indicating a certain degree of success in incorporating features of both instruments.

5. DISCUSSION

Transferring expressive affordances of the guitar, including string bending and strumming, to an extended keyboard interface proved successful in our tests with performers. Participants were in many cases able to explore familiar techniques in a new context while also finding new creative uses for these capabilities. Our experiment in transferring the guitar's constraints to the keyboard was less well-received. Most participants, even guitarists, did not see a reason that limitations on pitch range, simultaneous notes, or strings normally employed to strum a chord should be preserved in a keyboard interface. These results suggest that hybrid instrumental interfaces function best when combining the capabilities of all component models while minimizing limitations from any one instrument.

Future work will focus on aspects of continuous control, especially on achieving a better balance between pitch bend sensitivity and avoidance of uninentional slides. We will also explore improving the sonic realism of the string model. Finally, two users suggested that the interface could be useful in musical cultures (e.g. traditional Indian and Chinese music) where pitch sliding is a foundational technique. In general, we plan to explore how the principles of this interface can be used to make new intuitive, expressive instruments.

6. REFERENCES

[1] S. Gelineck and S. Serafin, "A practical approach towards an exploratory framework for physical modeling," *Computer Music Journal*, vol. 34, no. 2, pp. 51–65, 2010.

[2] A. McPherson and Y. Kim, "Design and applications of a multi-touch musical keyboard," in *Proc. SMC*, 2011.

[3] L. Fourier, "Jean-Jacques Perrey and the Ondioline," *Computer Music Journal*, vol. 18, no. 4, pp. 19–25, 1994.

[4] R. A. Moog and T. L. Rhea, "Evolution of the keyboard interface: The Bösendorfer 290 SE recording piano and the Moog multiply-touch-sensitive keyboards," *Computer Music Journal*, vol. 14, no. 2, pp. 52–60, Summer 1990.

[5] L. Haken, E. Tellman, and P. Wolfe, "An indiscrete music keyboard," *Computer Music Journal*, vol. 22, no. 1, pp. 30–48, 1998.

[6] R. Lamb and A. Robertson, "Seaboard: a new piano keyboard-related interface combining discrete and continuous control," in *Proc. ICMC*, 2011.

[7] P. Rothman, "The Ghost: an open-source, user programmable MIDI performance controller," in *Proc. NIME*, 2010.

[8] D. Schlessinger and J. O. Smith, "The Kalichord: A physically modeled electro-acoustic plucked string instrument," in *Proc. NIME*, 2009.

[9] M. Karjalainen, V. Välimäki, and T. Tolonen, "Plucked-string models: from the karplus-strong algorithm to digital waveguides and beyond," *Computer Music Journal*, vol. 22, no. 3, pp. 17–32, 1998.

[10] N. Lee, J. O. Smith, and V. Välimäki, "Analysis and synthesis of coupled vibrating strings using a hybrid modal-waveguide synthesis model," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 833–842, May 2010.

[11] J. Smith, J. Kuroda, J. Perng, and J. Abel, "Efficient computational modeling of piano strings for realtime synthesis using massspring chains, coupled finite differences, and digital waveguide sections," *Journal of the Acoustical Society of America*, vol. 128, no. 4, pp. 2344–2344, 2010.

[12] G. Evangelista and F. Eckerholm, "Player–instrument interaction models for digital waveguide synthesis of guitar: Touch and collisions," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 4, pp. 822–832, May 2010.

[13] J. Pakarinen, T. Puputti, and V. Välimäki, "Virtual slide guitar," *Computer Music Journal*, vol. 32, no. 3, pp. 42–54, 2008.