Virtual-Acoustic Instrument Design: Exploring the Parameter Space of a String-Plate Model

Sandor Mehes, Maarten van Walstijn, and Paul Stapleton
Sonic Arts Research Centre
School of Electronics, Electrical Engineering, and Computer Science
Queen's University Belfast, UK
{smehes01, m.vanwalstijn, p.stapleton}@qub.ac.uk

ABSTRACT

Exploration is an intrinsic element of designing and engaging with acoustic as well as digital musical instruments. This paper reports on the ongoing development of an explorative virtual-acoustic instrument based on simulation of the vibrations of a string coupled nonlinearly to a plate. The performer drives the model by tactile interaction with a string-board controller fitted with piezo-electric sensors. The string-plate model is formulated in a way that prioritises its parametric explorability. Where the roles of creating performance gestures and designing instruments are traditionally separated, such a design provides a continuum across these domains, with retainment of instrument physicality. The string-plate model, its real-time implementation, and the control interface are described, and the system is preliminarily evaluated through informal observations of how musicians engage with the system.

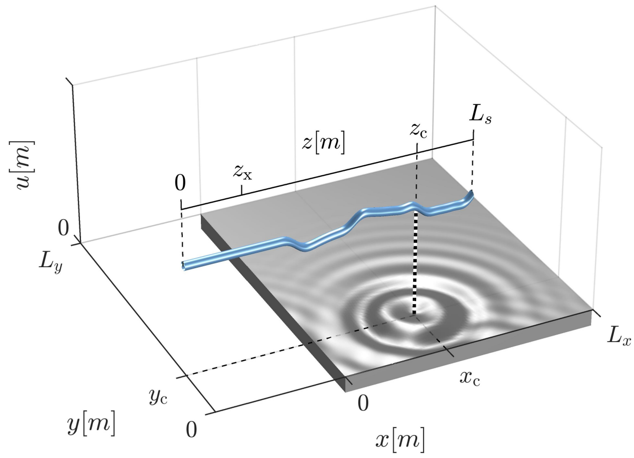

Figure 1: Geometry of the string-plate model.

Author Keywords

real-time physical modelling, parametric explorability, music improvisation, nonlinearity

1. INTRODUCTION

Among possible ways of mapping parameters to sound, physical modelling represents a special form that inherently constrains the sonic output to have a mechano-acoustic character, as such providing a rigorous basis for developing virtual-acoustic instruments. Such a computer-based system generally comprises a synthesis algorithm and a control interface [12], each subject to a number of design criteria, many of which overlap with those that the literature suggests apply more widely to digital musical instruments.

The control interface ideally instills a close coupling between the player and the instrument, mainly through aural and haptic feedback [11, 6]. Regarding the synthesis algorithm, a key feature of the physical modelling approach is that the oscillatory behaviour obeys a specified set of physical laws. It can be argued that certain associated benefits are retained when abstractions are introduced, such as allowing dimensional inhomogeneity in object connections [1, 3]. Nonetheless the design philosophy of the current authors is to preserve the Newtonian nature of the model as much as is feasible, as this provides the firmest basis for maintaining a sense of physical origin/source across a parametric sound domain. Ensuring such physicality is, however, by itself not that meaningful in a musical context; the instrument should also be engaging, afford sonic diversity, and encourage exploration [3]. As holds for other computer-based instruments [8], these aspects are generally better supported if the system features some form of nonlinear dynamics that gives rise to complex and surprising behaviours. In addition, physical models are often formulated in modular form, allowing the user to construct and explore new instruments by connecting elementary objects (see, e.g. [4, 2, 5, 16]). A further, lower-level requirement is that the algorithm is stable, robust, and accurate, as such approximating the underlying continuous-domain equations without significant artefacts. Finally, for use in live performance the algorithm's computational load must not exceed the real-time implementation limit. This requirement is more easily met if the algorithm can be scaled to the available hardware.

In practice, it can be difficult to meet all of these criteria, meaning that much of the initial design process tends to revolve around finding trade-offs that align well with the developer's design principles. The present study reports on the development of a virtual-acoustic instrument based on a string-plate model (see Fig. 1) that allows real-time adjustment of any of its parameters, as such prioritising *parametric explorability* over modularity. The instrument's versatility partly derives from the ability to generate sounds with harmonic (string-like) as well as inharmonic (plate- or beam-like) spectra and from the control over the level of coupling through adjusting the plate/spring mass ratio; its (modest) complexity arises from the nonlinearity of the spring coupling. The model is excited through an external force on the string. Following the controller design concept outlined in [7], a silent string controller is therefore

NIME'17, May 15-19, 2017, Aalborg University Copenhagen, Denmark.

employed as a physical interface that provides a tight and natural coupling with the synthesis algorithm.

Two specific objectives of the work are (1) to facilitate the best conditions for learning and navigating the physical model parameter space, and (2) to begin to investigate how musicians exploit such enhanced tunability. Full parametric explorability is not necessarily forthcoming in discrete-time modelling of distributed objects, usually due to the underlying grid form [16]. Following the approach taken in [14], the present study overcomes this by combining modal expansion with energy methods to formulate a gridless discrete-time model, the stability of which does not depend on any of the parameters.

The remainder of the paper is structured as follows. The string-plate model is summarised in Sect. 2. Sect. 3 specifies various details about the real-time implementation and control of the instrument, followed by initial observations of musicians engaging with the system in Sect. 4. Finally, Sect. 5 offers several concluding remarks and future perspectives.

2. STRING-PLATE MODEL

2.1 System Equations

Fig. 1 shows the geometry of the proposed model, in which the transverse vibrations of a stiff string are coupled to those of a thin rectangular plate via a spring element. The string is characterised by its length L_s, mass density ρ_s, cross-sectional area A_s, Young's modulus E_s, moment of inertia I_s. The plate is of dimensions $L_x \times L_y \times h_\mathrm{p}$, mass density ρ_p, and further characterised by the parameter $G_\mathrm{p} = (E_\mathrm{p} h_\mathrm{p}^3)/(12(1-\nu_\mathrm{p}^2))$, where ν_p is the Poisson ratio. Simply supported boundary conditions are imposed for both distributed objects. The dynamics of this system are governed by the equations

$$\rho_\mathrm{s} A_\mathrm{s} \frac{\partial^2 u_\mathrm{s}}{\partial t^2} = T_\mathrm{s} \frac{\partial^2 u_\mathrm{s}}{\partial z^2} - E_\mathrm{s} I_\mathrm{s} \frac{\partial^4 u_\mathrm{s}}{\partial z^4} - 2\rho_\mathrm{s} A_\mathrm{s} \zeta_\mathrm{s}(\beta) \frac{\partial u_\mathrm{s}}{\partial t} + \psi_\mathrm{c}(z) F_\mathrm{c}(t) + \psi_\mathrm{e}(z) F_\mathrm{e}(t), \tag{1}$$

$$\rho_\mathrm{p} h_\mathrm{p} \frac{\partial^2 u_\mathrm{p}}{\partial t^2} = -G_\mathrm{p} \Delta^2 u_\mathrm{p} - 2\rho_\mathrm{p} h_\mathrm{p} \zeta_\mathrm{p}(\beta) \frac{\partial u_\mathrm{p}}{\partial t} - \Psi_\mathrm{c}(x,y) F_\mathrm{c}(t), \tag{2}$$

where, for $\kappa = \mathrm{c}, \mathrm{e}$, the Dirac delta functions

$$\psi_\kappa(z) = \delta(z - z_\kappa) \tag{3}$$

specify point-like force distributions at the excitation position (z_e) and the connection position (z_c) along the string axis. Similarly,

$$\Psi(x,y) = \delta(x - x_\mathrm{c}, y - y_\mathrm{c}) \tag{4}$$

is a two-dimensional version of such a spatial distribution, with the spring connecting to the plate at coordinates $(x_\mathrm{c}, y_\mathrm{c})$.

The system is brought into vibration by the external force $F_\mathrm{e}(t)$, which excites the string. Frequency-dependent damping is effected by defining each of the decay rates ζ_κ ($\kappa = \mathrm{s}, \mathrm{p}$) as a function of the wave number:

$$\zeta_\kappa = \sigma_{\kappa,0} + \left(\sigma_{\kappa,1} + \sigma_{\kappa,3}\beta^2\right)|\beta|. \tag{5}$$

The spring connection force $F_\mathrm{c}(t)$ is defined as

$$F_\mathrm{c}(t) = k_\mathrm{L} u_\mathrm{c}(t) + k_\mathrm{N}\left[u_\mathrm{c}(t)\right]^3, \tag{6}$$

where k_L and k_N are stiffness-like coefficients, and where

$$u_\mathrm{c}(t) = u_\mathrm{p}(x_\mathrm{c}, y_\mathrm{c}, t) - u_\mathrm{s}(z_\mathrm{c}, t) \tag{7}$$

is the distance between the plate and string at the connection point. The cubic component in (6) allows simulation of stiffening springs, introducing nonlinear behaviour. Audio output signals are obtained by picking up the momentum $\rho_\mathrm{p} h_\mathrm{p} \frac{\partial u_\mathrm{p}}{\partial t}$ at K locations $(x_{\mathrm{a},k}, y_{\mathrm{a},k})$ on the plate.

2.2 Discretisation

Discretisation of the system in (1,2) is performed in similar fashion as for the tanpura model presented in [14]. That is, a modal expansion is applied to both the plate and the string equation, leading for each sub-system to a parallel set of modal oscillator equations of the form

$$m_z \frac{\partial^2 \bar{u}_{z,l}}{\partial t^2} = -k_{z,l} \bar{u}_{z,l}(t) - r_{z,i} \frac{\partial \bar{u}_{z,l}}{\partial t} + \bar{F}_{z,l}(t), \tag{8}$$

where, for $z = \mathrm{s}$ and $z = \mathrm{p}$ (referring to the string and the plate respectively), the constants m_z, r_z, and k_z are modal parameters which are calculated from the physical constants featuring in (1,2). The variables $\bar{u}_{z,l}$ and $\bar{q}_{z,l}$ represent the modal displacement and scaled momentum of the lth mode, respectively. Using a temporal step Δ_t, the first-order form of the modal oscillator equation in (8) is discretised by applying the finite-difference approximations

$$\frac{\partial u}{\partial t}\Big|_{t=(n+\frac{1}{2})\Delta_t} \approx \frac{\delta u}{\Delta_t}, \tag{9}$$

$$u\Big|_{t=(n+\frac{1}{2})\Delta_t} \approx \frac{\mu u}{2}, \tag{10}$$

which make use of the summing and difference operators

$$\mu u = u^{n+1} + u^n, \tag{11}$$

$$\delta u = u^{n+1} - u^n. \tag{12}$$

A correction is then applied to parameters of each mode to ensure exact modal frequencies and decay rates. For the system at hand, the spring connection force is discretised as

$$F_\mathrm{c}^{n+\frac{1}{2}} = \tfrac{1}{2} k_\mathrm{L} \mu u_\mathrm{c} + \tfrac{1}{4} k_\mathrm{N} \mu u_\mathrm{c} \mu u_\mathrm{c}^2, \tag{13}$$

which, given that we then have

$$F_\mathrm{c}^{n+\frac{1}{2}} = \frac{\delta V_\mathrm{c}}{\delta u_\mathrm{c}}, \tag{14}$$

where $V_\mathrm{c}(u_\mathrm{c}) = \frac{1}{2}u_\mathrm{c}^2 + \frac{1}{4}u_\mathrm{c}^4$ is the spring potential function, leads to a numerically conservative spring force and therefore an unconditionally stable formulation.

Similar to the derivation in [14], the discretised equations are combined in a vector-matrix update form. Here this involves two modal state column vectors for both the string and the plate:

$$\bar{\mathbf{u}}_\mathrm{s} = [\bar{u}_{\mathrm{s},1}, \bar{u}_{\mathrm{s},2}, \ldots, \bar{u}_{\mathrm{s},l}, \ldots, \bar{u}_{\mathrm{s},M_\mathrm{s}}],^\mathsf{T} \tag{15}$$

$$\bar{\mathbf{q}}_\mathrm{s} = [\bar{q}_{\mathrm{s},1}, \bar{q}_{\mathrm{s},2}, \ldots, \bar{q}_{\mathrm{s},l}, \ldots, \bar{q}_{\mathrm{s},M_\mathrm{s}}],^\mathsf{T} \tag{16}$$

$$\bar{\mathbf{u}}_\mathrm{p} = [\bar{u}_{\mathrm{p},1}, \bar{u}_{\mathrm{p},2}, \ldots, \bar{u}_{\mathrm{p},l}, \ldots, \bar{u}_{\mathrm{p},M_\mathrm{p}}],^\mathsf{T} \tag{17}$$

$$\bar{\mathbf{q}}_\mathrm{s} = [\bar{q}_{\mathrm{p},1}, \bar{q}_{\mathrm{p},2}, \ldots, \bar{q}_{\mathrm{p},l}, \ldots, \bar{q}_{\mathrm{p},M_\mathrm{p}}]^\mathsf{T}, \tag{18}$$

After performing several algebraic manipulations, it is found that the sample-by-sample update requires solving a cubic equation in s_c

$$g(s_\mathrm{c}) = c_3 s_\mathrm{c}^3 + c_2 s_\mathrm{c}^2 + c_1 s_\mathrm{c} + c_0, \tag{19}$$

where the coefficients are

$$c_0 = k_\mathrm{L}\phi_\mathrm{c} u_\mathrm{c}^n + k_\mathrm{N}\phi_\mathrm{c}(u_\mathrm{c}^n)^3 - e_\mathrm{c}, \tag{20}$$

$$c_2 = k_\mathrm{N}\phi_\mathrm{c} u_\mathrm{c}^n,$$

$$c_1 = \frac{1}{2}k_\mathrm{L}\phi_\mathrm{c} + \frac{3}{2}k_\mathrm{N}\phi_\mathrm{c}(u_\mathrm{c}^n)^2 + 1, \tag{21}$$

$$c_3 = \frac{1}{4}k_\mathrm{N}\phi_\mathrm{c}, \tag{22}$$

and where $\phi_\mathrm{c} > 0$ depends on several string and plate constants. Equation (19) is strictly monotonic and therefore has a unique solution; because $c_1 \geq 1$, the derivative of $g(s_\mathrm{c})$ is guaranteed to be at least unity, which helps avoiding large numbers of iterations. Hence it can be solved robustly at each time step using Newton's method (typically converging in less than 4 iterations).

The final algorithm can be summarised as follows. Knowing all the system variables at time n, the sample loop takes the form:

for $n = 1$ to N

$$F_{\mathrm{e}}^{n+\frac{1}{2}} = \tfrac{1}{2}\left(F_{\mathrm{e}}^{n+1} + F_{\mathrm{e}}^{n}\right) \rightarrow \text{update excitation force}$$

$$\bar{\mathbf{e}}_{\mathrm{s}} = \mathbf{c}_{\mathrm{s}} \odot \left[2\left(\bar{\mathbf{q}}_{\mathrm{s}}^{n} - \mathbf{a}_{\mathrm{s}} \odot \bar{\mathbf{u}}_{\mathrm{s}}^{n}\right)\right]$$

$$\bar{\mathbf{e}}_{\mathrm{p}} = \mathbf{c}_{\mathrm{p}} \odot \left[2\left(\bar{\mathbf{q}}_{\mathrm{p}}^{n} - \mathbf{a}_{\mathrm{p}} \odot \bar{\mathbf{u}}_{\mathrm{p}}^{n}\right)\right]$$

$$e_{\mathrm{c}} = \mathbf{g}_{\mathrm{p}}^{\mathsf{T}}\bar{\mathbf{e}}_{\mathrm{p}} - \mathbf{g}_{\mathrm{s}}^{\mathsf{T}}\bar{\mathbf{e}}_{\mathrm{s}} - \phi_{\mathrm{e}} F_{\mathrm{e}}^{n+\frac{1}{2}}$$

compute c_0, c_1, c_2 and solve (19)

$$F_{\mathrm{c}}^{n+\frac{1}{2}} = (e_{\mathrm{c}} - s_{\mathrm{c}})/\phi_{\mathrm{c}} \rightarrow \text{update spring force}$$

$$u_{\mathrm{c}}^{n+1} = s_{\mathrm{c}} + u_{\mathrm{c}}^{n} \rightarrow \text{update spring compression}$$

$$\bar{\mathbf{s}}_{\mathrm{s}} = \bar{\mathbf{e}}_{\mathrm{s}} + \mathbf{h}_{\mathrm{s}} F_{\mathrm{c}}^{n+\frac{1}{2}} + \mathbf{h}_{\mathrm{e}} F_{\mathrm{e}}^{n+\frac{1}{2}}$$

$$\bar{\mathbf{u}}_{\mathrm{s}}^{n+1} = \bar{\mathbf{s}}_{\mathrm{s}} + \bar{\mathbf{u}}_{\mathrm{s}}^{n} \rightarrow \text{update string modal displacements}$$

$$\bar{\mathbf{q}}_{\mathrm{s}}^{n+1} = \bar{\mathbf{s}}_{\mathrm{s}} - \bar{\mathbf{q}}_{\mathrm{s}}^{n} \rightarrow \text{update string modal momenta}$$

$$\bar{\mathbf{s}}_{\mathrm{p}} = \bar{\mathbf{e}}_{\mathrm{p}} - \mathbf{h}_{\mathrm{p}} F_{\mathrm{c}}^{n+\frac{1}{2}}$$

$$\bar{\mathbf{u}}_{\mathrm{p}}^{n+1} = \bar{\mathbf{s}}_{\mathrm{p}} + \bar{\mathbf{u}}_{\mathrm{s}}^{n} \rightarrow \text{update plate modal displacements}$$

$$\bar{\mathbf{q}}_{\mathrm{p}}^{n+1} = \bar{\mathbf{s}}_{\mathrm{p}} - \bar{\mathbf{q}}_{\mathrm{s}}^{n} \rightarrow \text{update plate modal momenta}$$

$$\mathbf{p}_{\mathrm{a}}^{n+1} = \mathbf{W}\mathbf{q}_{\mathrm{p}}^{n+1} \rightarrow \text{compute audio output signals}$$

end

with $\odot$ denoting elementwise multiplication, and where $\bar{\mathbf{u}}_{\mathrm{s}}^{n}$ and $\bar{\mathbf{q}}_{\mathrm{s}}^{n}$ are column vectors holding the M_{s} modal displacements and momenta of the string, respectively. Similarly, $\bar{\mathbf{u}}_{\mathrm{p}}^{n}$ and $\bar{\mathbf{q}}_{\mathrm{p}}^{n}$ represent the states of the M_{p} plate modes. The terms $(\mathbf{a}_{\mathrm{s}}, \mathbf{c}_{\mathrm{s}}, \mathbf{g}_{\mathrm{s}}, \mathbf{h}_{\mathrm{s}}, \mathbf{h}_{\mathrm{e}})$ and $(\mathbf{a}_{\mathrm{p}}, \mathbf{c}_{\mathrm{p}}, \mathbf{g}_{\mathrm{p}}, \mathbf{h}_{\mathrm{p}})$ are coefficient vectors depending on the physical parameters, and $\mathbf{W}$ is a $K \times M_{\mathrm{p}}$ matrix for computing the audio output signals.

2.3 User Parameters

Exposing the user to the full set of parameters featuring in (1,2) makes it unnecessarily difficult to learn navigating the parameter space because of parameter redundancy. Without loss of generality, the parameter set can be reduced by constraining the length parameters to $L_{\mathrm{s}} = L_x L_y = 1$ and fixing the string mass per unity length at $\rho_{\mathrm{s}} A_{\mathrm{s}} = 0.001$kg/m. The following parameters are then introduced with the aim of enabling intuitive navigation of the string-plate model parameter space:

$$\tilde{f}_{\mathrm{s}} = f_{\mathrm{s},1} = \frac{1}{2}\sqrt{\left(\frac{E_{\mathrm{s}} I_{\mathrm{s}}}{\rho_{\mathrm{s}} A_{\mathrm{s}}}\right)\pi^2 + \left(\frac{T_{\mathrm{s}}}{\rho_{\mathrm{s}} A_{\mathrm{s}}}\right)}, \tag{23}$$

$$\mathcal{B}_{\mathrm{s}} = \pi^2 \frac{E_{\mathrm{s}} I_{\mathrm{s}}}{T_{\mathrm{s}}}, \quad z_{\mathrm{c}}' = z_{\mathrm{c}}, \quad z_{\mathrm{e}}' = z_{\mathrm{e}}, \tag{24}$$

$$R_{\mathrm{ps}} = \frac{m_{\mathrm{p}}}{m_{\mathrm{s}}}, \quad R_{xy} = \frac{L_x}{L_y}, \tag{25}$$

$$\tilde{f}_{\mathrm{p}} = f_{\mathrm{p},1,1} = \frac{1}{2}\sqrt{\frac{G_{\mathrm{p}}\pi^2}{\rho_{\mathrm{p}} h_{\mathrm{p}}}}\left(L_x^{-2} + L_y^{-2}\right), \tag{26}$$

$$x_{\mathrm{c}}' = \frac{x_{\mathrm{c}}}{L_x}, \quad y_{\mathrm{c}}' = \frac{y_{\mathrm{c}}}{L_y}, \quad x_{\mathrm{a},k}' = \frac{x_{\mathrm{a},k}}{L_x}, \quad y_{\mathrm{a},k}' = \frac{y_{\mathrm{a},k}}{L_y}. \tag{27}$$

Furthermore, the connection spring stiffness constants in (6) are parameterised as follows:

$$k_{\mathrm{L}} = (1-\eta)k_{\mathrm{c}}, \quad k_{\mathrm{N}} = \eta\, k_{\mathrm{c}} \cdot 10^8, \tag{28}$$

where k_{c} is an overall stiffness parameter and η gives control over the level of nonlinear behaviour.

```
INITIAL STATE OF SWITCHES
t paramSwitch = 0;
t paramSubSwitch = 0;

id parameUpdateFunc() {
//LEVEL1
if (parameter != prevParam) {
  prevParam = parameter;
  paramSwitch = 1;
}
//LEVEL2
if (paramSwitch == 1) {
  paramSub = calculate new variables;
  paramSubSwitch = 1;
}
//RESET SWITCHES
paramSwitch = 0;
paramSubSwitch = 0;
```

Figure 2: Parameter update function example.

3. REALISATION

3.1 Real-Time Synthesis within Audio Unit

The model described in the previous section first was implemented in Matlab, with one second of simulation requiring about three seconds to compute. For real-time rendering, the system was built in Audio Unit plug-in architecture, coding in C++ within the JUCE application framework [10]. This yields executable code within a standard plug-in API provided by Core Audio. Running at 48 kHz, A 120 sample buffersize was used, thus effecting a 400 Hz parameter control rate. On our machine the algorithm runs without underflow for a maximum total of about 2000 modes.

To optimise computations that loop over the string and plate modes, `FloatVectorOperations` were employed. This class utilises Apple's vDSP functions[1], performing the same operation on multiple modes simultaneously to minimise any overheads. Further efficiency savings were made regarding the calculation of the system coefficients from time-varying parameters, which was implemented in layered fashion. That is, rather than updating all system coefficients at control rate, each individual parameter change triggers only the smallest set of recalculations at lower levels. At each level, the current and the previous values of a constant are compared, and any dependent constants at lower levels stay idle unless these values are different; a binary switch is defined for each coefficient, which keeps track of whether that coefficient has changed or not. The process continues to the lowest level, where the updated coefficients are passed to the main loop, after which all switches are reset to 0. Fig. 2 illustrates this process for a case of two levels and two parameters; the actual coefficient update function comprises ten levels, twenty-two coefficient vectors, and twelve scalars coefficients.

3.2 Physical Control Interfaces

A new string-board controller (see Fig. 3) was designed and built, based on the concept described in detail in [7], which provides a calibrated signal estimation of the forces excerted by the musician. In this new version, each of the bridges consists of a plastic cylindrical piston housed by a metal holder, as such sensing forces only in the vertical direction. The calibration, which removes the 'nasal' timbre normally observed with piezo disks, is now effected by scaling the modal exciatation weights in the string model, rather than using a dedicated digital filter at the input stage. Although initially intended to be played through plucking, scraping, tapping, and bowing the string, force signals can also be generated by tapping the board it is mounted on; this alter-

[1] See `https://developer.apple.com/library/content/documentation/Performance/Conceptual/vDSP_Programming_Guide`

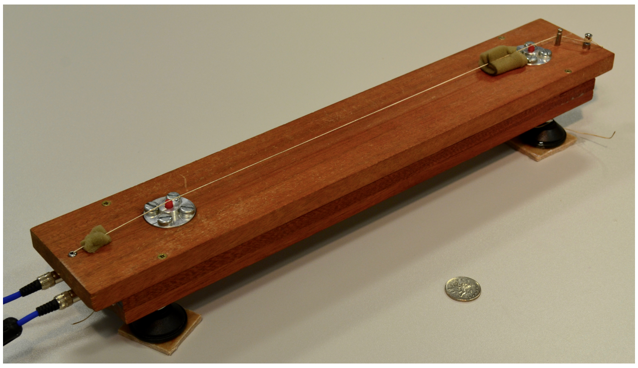

Figure 3: The string-board controller. The string is stretched over two bridge pistons, each of which rests on an internally mounted piezoelectric disk that sense the vibrations. Foam pieces are employed to suppress round-trip waves along the string.

native mode of interaction affords slightly different nuances in the generated excitation signals.

To provide fine-grained tactile control of the string-plate parameters, a Knobbee 32 was employed [9], which supports OSC messages at 10 bit resolution. An audio-visual impression of the string-plate model, the controller setup and the kind of sounds that can be produced with the system is provided on the companion webpage[2].

4. INFORMAL OBSERVATIONS OF MUSICIANS ENGAGING WITH THE SYSTEM

The system was made available for a few hours to four musicians of varying background and experience, each of whom produced a short piece of music. The main purpose was to obtain preliminary insight into their decision making regarding parameter control. The plug-in was hosted by a digital audio workstation. This allowed musicians to drive the model with either pre-stored signals or by live output from the string-board controller, which can be excited either on the string (plucking or bowing) or the board (tapping, drumming). Within this exercise, the following informal observations were made:

- All four musicians appeared to make distinctions between 'design' and 'gesture', in that some parameters were fine-tuned and left unchanged thereafter or changed infrequently, while other parameters where continuously updated for one or more periods during performance.
- We observed different approaches to making such functional distinctions. One of the musicians, who took a compositional approach, used the bulk of the parameters in a design manner, with the gestural focus on f_s, f_p, R_{ps} and the listening positions. This contrasted with how one of the other musicians created gestural passages via gradual changes in more than half of the parameters, while treating all position parameters except z'_c and z'_e largely as design parameters.
- Across the four musicians, the parameters f_s, f_p, R_{xy}, k_c, and z'_c were used mostly in a gestural sense, while $\sigma_{s,0}$, $\sigma_{s,3}$, $\sigma_{p,0}$, x'_c, y'_c and η were treated largely as design parameters. The remaining parameters were used in both ways, in several cases by the same musician.
- Unsurprisingly, some of the more dramatic gestures were effected by varying the frequency parameters f_s and f_p. A less expected observation was that the musicians also made extensive use of R_{ps}, R_{xy}, and k_c to articulate the sonic output.
- A common element in the musician's feedback to us was that individual knob control of each of the parameters is very suited to design tasks, for gestural tasks they would prefer to control groups of parameters.
- Two musicians mentioned that the gestural control can be further improved by integrating the knobs and the string controller into a single physical interface.

[2] `www.socasites.qub.ac.uk/mvanwalstijn/nime17`

5. CONCLUSIONS

The real-time string-plate model allows on-line adjustment of any of its parameters with instant aural feedback, as such inviting the user to interactively engage with it in (1) a design sense, by serendipitously exploring the parameter space, and (2) a gestural sense, through articulatory parameter adjustments during performance. Regarding the model itself, its physical nature supports intuitive material and geometric orientation, and the nonlinearity of the spring connection adds a richness and complexity to the parameter space that allows for the emergence of new behaviours and unexpected and engaging musical results. The string-board facilitates effortful, continuous excitation control, interfacing between the musician and the model. The virtual string-plate acts as a non-transparent, vibro-acoustic mediator of the tactile detail enacted by the performer. These system affordances are well aligned with the practices and idiosyncrasies of live performance and music improvisation [13, 15].

Although the musicians who experimented with the system took rather diverse approaches and decisions in assigning and exploring these functionalities, some commonalities were observed. In particular, some agreement appears to exist on which parameters are most suited to creating gestures. Interestingly, none of the four users created any sounds with the nonlinearity level parameter η set lower than 0.9, suggesting that nonlinear behaviour is indeed a key synthesis ingredient of any virtual-acoustic instrument.

Among possible improvements and extensions, an obvious one is further code optimisation so that the system can be rendered with the string and plate mode frequencies covering the whole of the hearing bandwidth for 'larger' strings and plates. The authors are currently also developing an extended version of the synthesis algorithm that incorporates a different type of nonlinearity in the string-plate connection.

Further future research avenues include devising new strategies for gestural control of the parameters and development of alternatives to and refinements of the string-board controller. Progress on these points inevitably involves close interaction between engineers and musicians, which will be sustained in future work in the form of both formal user studies and performative dissemination.

6. REFERENCES

[1] G. Essl. The Displaced Bow and APhISMs: Abstract Physically Informed Synthesis Methods for Composition and Interactive Performance. In *Florida Electronic Music Festival, University of Florida*, Gainsville, Florida, 2003.

[2] S. Gelineck and S. Serafin. Phoxes - Modular Electronic Music Instruments Based on Physical Modeling Sound Synthesis. In *7th Sound and Music Computing Conference*, 2010.

[3] S. Gelineck and S. Serafin. A Practical Approach Towards an Exploratory Framework for Physical Modeling. *Computer Music Journal*, 34(2):51–65, 2010.

[4] D. M. Howard and S. Rimell. Real-Time Gesture-Controlled Physical Modelling Music Synthesis with Tactile Feedback. *EURASIP Journal on Advances in Signal Processing*, 2004(7):1001–1006, 2004.

[5] J. Leonard and C. Cadoz. Physical Modelling Concepts for a Collection of Multisensory Virtual Musical Instruments. In *New Interfaces for Musical Expression 2015*, pages 150–155, 2015.

[6] A. Luciani, J.-L. Florens, D. Couroussé, and C. Cadoz. Ergotic Sounds: A New Way to Improve Playability, Believability and Presence of Digital Musical Instruments. In *ENACTIVE/07-International Conference on Enactive Interfaces*, pages 373–376. ACROE, 2007.

[7] S. Mehes, M. van Walstijn, and P. Stapleton. Towards a Virtual-Acoustic String Instrument. In *13th Sound and Music Computing Conference*, pages 286–292, Hamburg, Germany, 2016.

[8] T. Mudd, S. Holland, P. Mulholland, and N. Dalton. Investigating the Effects of Introducing Nonlinear Dynamical Processes into Digital Musical Interfaces. In *12th Sound and Music Computing Conference*, pages 15–21, Maynooth, Ireland, 2015.

[9] C. Popp and R. Soria-Luz. Developing Mixer-style Controllers Based On Arduino/Teensy Microcontrollers. In *The 12th Sound and Music Computing Conference*, pages 37–41, Maynooth, Ireland, 2015.

[10] M. Robinson. *Getting Started with JUCE*. Packt Publishing Ltd., Birmingham, UK, 1st edition, 2013.

[11] S. Serafin, M. Burtner, S. Nichols, and S. O'Modhrain. Expressive Controllers for Bowed String Physical Models. In *Proceedings of the 19th International Conference on Digital Audio Effects*, Limerick, Ireland, 2001.

[12] J. O. Smith. Virtual Acoustic Musical Instruments: Review and Update. *Journal of New Music Research*, 33(3):283–304, 2004.

[13] P. Stapleton, S. Waters, N. Ward, and O. Green. Distributed Agency in Performance. In *Proceedings of the International Conference on Live Interfaces (ICLI2016)*, pages 329–330, Brighton, UK, 2016.

[14] M. van Walstijn, J. Bridges, and S. Mehes. A Real-Time Synthesis Oriented Tanpura Model. In *Proceedings of the 19th International Conference on Digital Audio Effects*, pages 175–182, Brno, Czech Republic, 2016.

[15] S. Waters. Touching at a Distance: Resistance, Tactility, Proxemics and the Development of a Hybrid Virtual/Physical Performance System. *Contemporary Music Review*, 32(2):119–134, 2013.

[16] C. J. Webb and S. Bilbao. On the Limits of Real-Time Physical Modelling Synthesis with a Modular Environment. In *Proceedings of the 18th International Conference on Digital Audio Effects*, pages 65–72, Trondheim, Norway, 2015.