AN EXPLORATIVE STRING-BRIDGE-PLATE MODEL WITH TUNABLE PARAMETERS

Maarten van Walstijn, Sandor Mehes

Sonic Arts Research Centre
School of Electronics, Electrical Engineering, and Computer Science
Queen's University Belfast, UK
{m.vanwalstijn,smehes01}@qub.ac.uk

ABSTRACT

The virtual exploration of the domain of mechano-acoustically produced sound and music is a long-held aspiration of physical modelling. A physics-based algorithm developed for this purpose combined with an interface can be referred to as a virtual-acoustic instrument; its design, formulation, implementation, and control are subject to a mix of technical and aesthetic criteria, including sonic complexity, versatility, modal accuracy, and computational efficiency. This paper reports on the development of one such system, based on simulating the vibrations of a string and a plate coupled via a (nonlinear) bridge element. Attention is given to formulating and implementing the numerical algorithm such that any of its parameters can be adjusted in real-time, thus facilitating musician-friendly exploration of the parameter space and offering novel possibilities regarding gestural control. Simulation results are presented exemplifying the sonic potential of the string-bridge-plate model (including bridge rattling and buzzing), and details regarding efficiency, real-time implementation and control interface development are discussed.

1. INTRODUCTION

Physical modelling studies often focus on the simulation and virtualisation of a specific musical instrument or class thereof [1]. Progress in this field has, however, also been driven by the prospect of new variations of virtual-acoustic instruments. Going beyond the aim of faithful imitation, the challenge then shifts towards design and control, seeking physical model configurations, implementations, and user interfaces that facilitate creative exploration of the domain of mechanically/acoustically plausible sounds. Notable past efforts in this area include several early physical modelling software environments such as CORDIS [2], MOSAIC [3], and TAO [4]. These systems and more recent variations of the same concept (see, e.g. [5, 6, 7, 8]) facilitate the construction of new instruments by connecting either elementary masses or distributed objects (e.g. strings, membranes), usually via spring elements.

Beyond such modularity, the user invariably faces the task of learning to navigate the parameter space of the specified configuration. Even though the parameters are physical and therefore intuitive, this tends to be a formidable exercise for all but the simplest systems, especially so if it has to be performed off-line. A principal motivation behind the present study is to devise virtual-acoustic instruments that facilitate this learning and exploration process. This is pursued here by developing a specific physical configuration (hence de-emphasising modularity) that allows the user to perform design and control tasks via on-line tuning of any of the model parameters with instant aural feedback.

The first challenge that arises from this objective is one of design, i.e. determining what kind of physical model configuration is appropriate as a testbed. Drawing inspiration from several relevant DAFx studies [9, 5, 8, 10] and partly building on earlier ideas [11, 12], the proposed model takes the form of a string and a plate connected by a parameterised bridge element, with a local damper fitted on the string (see Fig. 1). The bridge can be parametrically configured to simulate different types of linear and nonlinear coupling, including mass-like behaviour, spring stiffening and contact phenomena (i.e. rattling and buzzing).

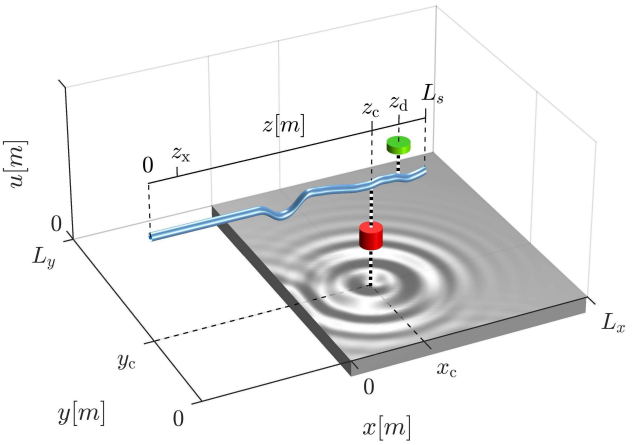

Figure 1: *Geometry of the string-bridge-plate model. The red cylinder represents the bridge mass, and the green disk indicates where the string damper is located.*

The second, more technical challenge - the addressing of which the bulk of this paper is devoted to - consists in the derivation of a computationally robust and sufficiently efficient numerical formulation that supports on-line parameter adjustment, and can be scaled to common hardware for real-time operation. The approach taken to address this can be summarised as follows. The system equations of the string-bridge-plate model are set out in § 2. An unconditionally stable and modally exact numerical formulation is then developed in § 3. This avoids complications and restrictions regarding on-line variation of some of the model parameters that would arise with more general spatio-temporal discretisation. Similar to the Port-Hamiltonian approach [13], discretisation is performed over a first-order form, which in the presence of non-smooth forces can help avoid spurious oscillations [14]. The proposed model is computationally robust due to (a) its provable stability, (b) the provable uniqueness and demonstrable convergence of the solution to the nonlinear equation that has to be found iteratively at each discrete-time instant, and (c) the empirically set constraints on each of the tunable parameters. The latter is needed

to limit the number of required nonlinear solver iterations. The model's sonic potential is exemplified with numerical experiments in § 4, which also reports on real-time implementation aspects and on preliminary findings with the initial control configuration. Finally, § 5 offers concluding remarks and future perspectives.

2. STRING-BRIDGE-PLATE MODEL

2.1. System Equations

Consider transverse vibrations (u) of a stiff string and a thin rectangular plate, both under simply supported boundary conditions, coupled via a bridge element of a mass m_b with parameterised spring elements, and with local damping applied at $z = z_\mathrm{d}$ along the string axis (see Fig. 1). 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, and damping factor γ_s of yet to be determined wave-number dependency. The plate is of dimensions $L_x \times L_y \times h_\mathrm{p}$, mass density ρ_p, and experiences damping according to γ_p. Two of its material properties are encapsulated in 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. The dynamics of the coupled system are governed by

$$\rho_\mathrm{s} A_\mathrm{s} \frac{\partial^2 u_\mathrm{s}}{\partial t^2} = T_\mathrm{s} \frac{\partial^2 u_\mathrm{s}}{\partial z^2} - E_\mathrm{s} I_\mathrm{s} \frac{\partial^4 u_\mathrm{s}}{\partial z^4} - \gamma_\mathrm{s} \frac{\partial u_\mathrm{s}}{\partial t} + \psi_\mathrm{c}(z) F_1(t) + \psi_\mathrm{x}(z) F_\mathrm{x}(t) + \psi_\mathrm{d}(z) F_\mathrm{d}(t), \quad (1)$$

$$m_\mathrm{b} \frac{\partial^2 u_\mathrm{b}}{\partial t^2} = -r_\mathrm{b} \frac{\partial u_\mathrm{b}}{\partial t} - F_1(t) + F_2(t) + F_\mathrm{b}(t), \quad (2)$$

$$\rho_\mathrm{p} h_\mathrm{p} \frac{\partial^2 u_\mathrm{p}}{\partial t^2} = -G_\mathrm{p} \nabla^4 u_\mathrm{p} - \gamma_\mathrm{p} \frac{\partial u_\mathrm{p}}{\partial t} - \Psi_\mathrm{c}(x,y) F_2(t), \quad (3)$$

where, for $\kappa = \mathrm{c}, \mathrm{x}, \mathrm{d}$, $\psi_\kappa(z) = \delta(z - z_\kappa)$ and $\Psi(x,y) = \delta(x - x_\mathrm{c}, y - y_\mathrm{c})$ are single-point spatial distributions, and $\nabla^4 = (\partial^4/\partial x^4 + \partial^4/\partial y^4)$ is a biharmonic operator. It is readily seen that (1) models a beam rather than a string when $T_\mathrm{s} \ll E_\mathrm{s} I_\mathrm{s}$.

The system is brought into vibration by $F_\mathrm{x}(t)$, which excites the string, and/or $F_\mathrm{b}(t)$ which drives the bridge mass; bridge damping is controlled with r_b. The damper force - which allows suppression of string oscillations on either side of the connection point - is $F_\mathrm{d}(t) = -r_\mathrm{d} \frac{\partial u_\mathrm{d}}{\partial t}$, where $u_\mathrm{d}(t) = u_\mathrm{s}(z_\mathrm{d}, t)$. The spring connection forces F_1 and F_2 are functions of the inter-object distances $u_1(t) = u_\mathrm{b}(t) - u_\mathrm{s}(z_\mathrm{c}, t)$ and $u_2(t) = u_\mathrm{p}(x_\mathrm{c}, y_\mathrm{c}, t) - u_\mathrm{b}(t)$ as follows ($\ell = 1, 2$):

$$F_\ell(t) = k_\mathrm{L} u_\ell(t) + k_\ell^+ \lfloor u_\ell(t) \rfloor^\alpha - k_\ell^- \lfloor -u_\ell(t) \rfloor^\alpha, \quad (4)$$

where $\lfloor u \rfloor \triangleq \max(0, u)$, k_L, k_ℓ^+, and k_ℓ^- are stiffness coefficients, and $\alpha \geq 1$ is a power law exponent. The specific form of (4) includes various types of linear and nonlinear restoring-force based connections, allowing to parametrically remove or add the pushing or pulling force of each of the springs through adjusting k_ℓ^+, and k_ℓ^-, respectively; Fig. 2 shows a few example force-distance curves. Modelling of the bridge in this manner facilitates the simulation of a variety of connection configurations, four example cases of which are shown in Fig. 3. Note that while the bridge mass cannot be set to zero in our model, it can nonetheless be made negligible by making it very small compared with the string mass. For audio ouput, we define the plate momentum at K pickup positions $(x_{\mathrm{a},k}, y_{\mathrm{a},k})$:

$$p_{\mathrm{a},k}(t) = \rho_\mathrm{p} h_\mathrm{p} \frac{\partial}{\partial t} u_\mathrm{p}(x_{\mathrm{a},k}, y_{\mathrm{a},k}, t). \quad (5)$$

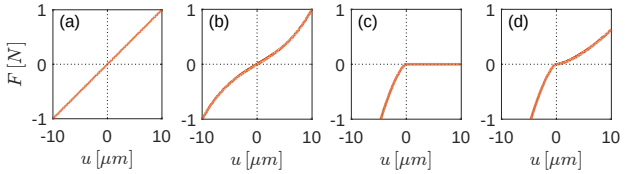

Figure 2: *Examples of spring force-distance curves. (a): linear spring [$k_\mathrm{L} = 10^5$, $k_\ell^\pm = 0$]. (b): stiffening spring [$k_\mathrm{L} = 6 \times 10^4$, $k_\ell^+ = k_\ell^- = 4 \times 10^{14}$, $\alpha = 3$]. (c) one-sided spring [$k_\mathrm{L} = 0$, $k_\ell^+ = 0$, $k_\ell^- = 1 \times 10^8$, $\alpha = 1.5$] (d): asymmetric spring [$k_\mathrm{L} = 0$, $k_\ell^+ = 1 \times 10^8$, $k_\ell^+ = 2 \times 10^7$, $\alpha = 1.5$].*

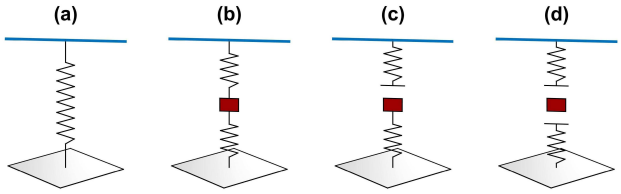

Figure 3: *Bridge configuration examples. (a): massless bridge [$m_\mathrm{b} \ll \rho_\mathrm{s} A_\mathrm{s} L_\mathrm{s}$, $k_1^\pm = k_2^\pm$]. (b): mass-spring bridge [$k_1^\pm = k_2^\pm$]. (c): 'flat' bridge [$k_\mathrm{L} = 0$, $k_1^+, k_2^+, k_2^- > 0$, $k_1^- = 0$]. (d): rattling bridge [$k_\mathrm{L} = 0$, $k_1^+, k_2^+ > 0$, $k_1^- = k_2^- = 0$].*

2.2. Modal Expansion

The displacement of each of the two distributed linear sub-systems can be written as a modal expansion:

$$u_\mathrm{s}(z,t) = \sum_{i=1}^{M_\mathrm{s}} v_{\mathrm{s},i}(z)\, \bar{u}_{\mathrm{s},i}(t), \quad (6)$$

$$u_\mathrm{p}(x,y,t) = \sum_{i=1}^{M_\mathrm{x}} \sum_{j=1}^{M_\mathrm{y}} v_{\mathrm{p},i,j}(x,y)\, \bar{u}_{\mathrm{p},i,j}(t), \quad (7)$$

where

$$v_{\mathrm{s},i} = \sin(\beta_i x), \quad v_{\mathrm{p},i,j}(x,y) = \sin(\beta_{\mathrm{x},i} x) \sin(\beta_{\mathrm{y},j} y), \quad (8)$$

are the respective mode shape functions under simply supported boundary conditions. The wave numbers are $\beta_i = i\pi/L_\mathrm{s}$ for the string/beam and $\beta_{\mathrm{x},i} = i\pi/L_\mathrm{x}, \beta_{\mathrm{y},j} = j\pi/L_\mathrm{y}$ for the plate (the overall wave number defined as $\beta_{i,j} = \sqrt{\beta_{\mathrm{x},i}^2 + \beta_{\mathrm{y},i}^2}$). For both the string ($\kappa = \mathrm{s}$) and the plate ($\kappa = \mathrm{p}$), the modal displacements ($\bar{u}_\kappa$) in (6) and (7) are governed by second-order differential equations of the form

$$m_\kappa \frac{\partial^2 \bar{u}_{\kappa,l}}{\partial t^2} = -k_{\kappa,l} \bar{u}_{\kappa,l}(t) - r_{\kappa,i} \frac{\partial \bar{u}_{\kappa,l}}{\partial t} + \bar{F}_{\kappa,l}(t), \quad (9)$$

where we have introduced a new modal index l, which maps as $l = i$ for the string and as $l = (M_y - 1)i + j$ for the plate, and where the respective modal parameters are

$$m_\mathrm{s} = \frac{\rho_\mathrm{s} A_\mathrm{s} L_\mathrm{s}}{2}, \; k_{\mathrm{s},l} = \frac{L_\mathrm{s}}{2} \left(E_\mathrm{s} I_\mathrm{s} \beta_l^4 + T_\mathrm{s} \beta_l^2 \right), \; r_{\mathrm{s},l} = \frac{L_\mathrm{s}}{2} \gamma_\mathrm{s}(\beta_l), \quad (10)$$

$$m_\mathrm{p} = \frac{\rho_\mathrm{p} h_\mathrm{p} L_\mathrm{x} L_\mathrm{y}}{4}, \; k_{\mathrm{p},l} = \frac{L_\mathrm{x} L_\mathrm{y} G_\mathrm{p}}{4} \beta_l^4, \; r_{\mathrm{p},l} = \frac{L_\mathrm{x} L_\mathrm{y}}{4} \gamma_\mathrm{p}(\beta_l), \quad (11)$$

The modal frequencies are $\omega_{\kappa,l} = \sqrt{k_{\kappa,l}/m_\kappa - \zeta_{\kappa,l}^2}$, and the modal attenuation rates are parameterised here in the convenient low-parameter frequency-dependent form

$$\zeta_{\kappa,\ell} = \frac{r_\kappa}{2m_\kappa} = \sigma_{\kappa,0} + \sigma_{\kappa,1}\beta_\ell + \sigma_{\kappa,3}\beta_\ell^3, \tag{12}$$

which retrospectively defines the damping parameters γ_s and γ_p in equations (1) and (3) as wave-number dependent. For the two sub-systems, the modal force term in (9) is

$$\begin{aligned}\bar{F}_{\mathrm{s},l}(t) &= \int_0^{L_\mathrm{s}} v_{\mathrm{s},l}(z)\Big[\psi_\mathrm{c}(z)F_1(t) + \psi_\mathrm{x}(z)F_\mathrm{x}(t) + \psi_\mathrm{d}(z)F_\mathrm{d}(t)\Big]dz \\ &= g_{\mathrm{s},l}F_1(t) + g_{\mathrm{x},l}F_\mathrm{x}(t) + g_{\mathrm{d},l}F_\mathrm{d}(t), \end{aligned}\tag{13}$$

$$\begin{aligned}\bar{F}_{\mathrm{p},l}(t) &= -\int_0^{L_\mathrm{x}}\int_0^{L_\mathrm{y}} v_{\mathrm{p},l}(x,y)\psi(x,y)F_2(t)dydx \\ &= -g_{\mathrm{p},l}F_2(t), \end{aligned}\tag{14}$$

where (for $\kappa = \mathrm{e}, \mathrm{d}$)

$$g_{\mathrm{s},l} = v_{\mathrm{s},l}(z_\mathrm{c}), \quad g_{\kappa,l} = v_{\mathrm{s},l}(z_\kappa), \quad g_{\mathrm{p},l} = v_{\mathrm{p},l}(x_\mathrm{c}, y_\mathrm{c}). \tag{15}$$

For audio output, the plate momenta are modally expanded as

$$p_{\mathrm{a},k} = \rho_\mathrm{p} h_\mathrm{p} \sum_{l=1}^{M_\mathrm{p}} g_{\mathrm{a},k,l}\,\frac{\partial \bar{u}_{\mathrm{p},l}}{\partial t}, \tag{16}$$

where $g_{\mathrm{a},k,l} = v_{\mathrm{p},l}(x_{\mathrm{a},k}, y_{\mathrm{a},k})$ and $M_\mathrm{p} = M_x M_y$.

3. NUMERICAL FORMULATION

3.1. Discretisation in Time

The differential equations (9) and (2) are first re-written in first-order form:

$$\begin{aligned}\frac{\partial \bar{u}_{\mathrm{s},l}}{\partial t} &= \frac{\bar{p}_{\mathrm{s},l}}{m_\mathrm{s}}, \quad \frac{\partial \bar{p}_{\mathrm{s},l}}{\partial t} = -k_{\mathrm{s},l}\bar{u}_{\mathrm{s},l} - r_{\mathrm{s},l}\frac{\partial \bar{u}_{\mathrm{s},l}}{\partial t} \\ &\quad + g_{\mathrm{s},l}F_1 + g_{\mathrm{x},l}F_\mathrm{x} + g_{\mathrm{d},l}F_\mathrm{d}, \end{aligned}\tag{17}$$

$$\frac{\partial u_\mathrm{b}}{\partial t} = \frac{p_\mathrm{b}}{m_\mathrm{b}}, \quad \frac{\partial p_\mathrm{b}}{\partial t} = -r_\mathrm{b}\frac{\partial u_\mathrm{b}}{\partial t} - F_1 + F_2 + F_\mathrm{b}, \tag{18}$$

$$\frac{\partial \bar{u}_{\mathrm{p},l}}{\partial t} = \frac{\bar{p}_{\mathrm{p},l}}{m_\mathrm{p}}, \quad \frac{\partial \bar{p}_{\mathrm{p},l}}{\partial t} = -k_{\mathrm{p},l}\bar{u}_{\mathrm{p},l} - r_{\mathrm{p},l}\frac{\partial \bar{u}_{\mathrm{p},l}}{\partial t} - g_{\mathrm{p},l}F_2, \tag{19}$$

where p_κ denotes momentum. Gridding time as $u^n \mathrel{\hat{=}} u(n\Delta_t)$, the following *sum* and *difference* operators

$$\mu u = u^{n+1} + u^n, \quad \delta u = u^{n+1} - u^n, \tag{20}$$

are then employed to discretise these equations at $t = (n+\frac{1}{2})\Delta_t$, which (with $F_\mathrm{x}^{n+\frac{1}{2}} = \frac{1}{2}\mu F_\mathrm{x}$, $F_\mathrm{b}^{n+\frac{1}{2}} = \frac{1}{2}\mu F_\mathrm{b}$) yields

$$\begin{aligned}\frac{\delta \bar{u}_{\mathrm{s},l}}{\Delta_t} &= \frac{\mu \bar{p}_{\mathrm{s},l}}{2m_\mathrm{s}}, \quad \frac{\delta \bar{p}_{\mathrm{s},l}}{\Delta_t} = -k^*_{\mathrm{s},l}\frac{\mu \bar{u}_{\mathrm{s},l}}{2} - r^*_{\mathrm{s},l}\frac{\delta \bar{u}_{\mathrm{s},l}}{\Delta_t} \\ &\quad + g^*_{\mathrm{s},l}F_1^{n+\frac{1}{2}} + g^*_{\mathrm{x},l}F_\mathrm{x}^{n+\frac{1}{2}} + g^*_{\mathrm{d},l}F_\mathrm{d}^{n+\frac{1}{2}}, \end{aligned}\tag{21}$$

$$\frac{\delta u_\mathrm{b}}{\Delta_t} = \frac{\mu p_\mathrm{b}}{2m_\mathrm{b}}, \quad \frac{\delta p_\mathrm{b}}{\Delta_t} = -r_\mathrm{b}\frac{\delta u_\mathrm{b}}{\Delta_t} - F_1^{n+\frac{1}{2}} + F_2^{n+\frac{1}{2}} + F_\mathrm{b}^{n+\frac{1}{2}}, \tag{22}$$

$$\frac{\delta \bar{u}_{\mathrm{p},l}}{\Delta_t} = \frac{\mu \bar{p}_{\mathrm{p},l}}{2m_\mathrm{p}}, \quad \frac{\delta \bar{p}_{\mathrm{p},l}}{\Delta_t} = -k^*_{\mathrm{p},l}\frac{\mu \bar{u}_{\mathrm{p},l}}{2} - r^*_{\mathrm{p},l}\frac{\delta \bar{u}_{\mathrm{p},l}}{\Delta_t} - g^*_{\mathrm{p},l}F_2^{n+\frac{1}{2}}. \tag{23}$$

The damper and spring connection forces ($\ell = 1, 2$) are discretised as

$$F_\mathrm{d}^{n+\frac{1}{2}} = -r_\mathrm{d}\frac{\delta u_\mathrm{d}}{\Delta_t}, \quad F_\ell^{n+\frac{1}{2}} = k_\mathrm{L}\frac{\mu u_\ell}{2} + \frac{\delta V_\ell}{\delta u_\ell}, \tag{24}$$

the latter utilising the nonlinear spring element potentials

$$V_\ell(u_\ell) = \left(\frac{k_\ell^+}{\alpha+1}\right)\lfloor u_\ell \rfloor^{\alpha+1} + \left(\frac{k_\ell^-}{\alpha+1}\right)\lfloor -u_\ell \rfloor^{\alpha+1}. \tag{25}$$

Note that in (21) and (23) the modal elasticity and damping constants of the string and plate have been substituted as follows:

$$k_{\kappa,l} \to k^*_{\kappa,l} = \frac{4m_\kappa a^*_{\kappa,l}}{\Delta_t^2}, \quad r_{\kappa,l} \to r^*_{\kappa,l} = \frac{2m_\kappa b^*_{\kappa,l}}{\Delta_t}. \tag{26}$$

where, with $R_{\kappa,l} = \exp(-\zeta_{\kappa,l}\Delta_t)$ and $\Omega_{\kappa,l} = \cos(\omega_{\kappa,l}\Delta t)$,

$$a^*_{\kappa,l} = \frac{1 - 2R_{\kappa,l}\Omega_{\kappa,l} + R^2_{\kappa,l}}{1 + 2R_{\kappa,l}\Omega_{\kappa,l} + R^2_{\kappa,l}}, \quad b^*_{\kappa,l} = \frac{2\left(1 - R^2_{\kappa,l}\right)}{1 + 2R_{\kappa,l}\Omega_{\kappa,l} + R^2_{\kappa,l}}. \tag{27}$$

As explained in previous work [12], this eliminates numerical dispersion and attenuation at the sub-system resonance frequencies. In addition, each modal weight $g_{\kappa,l}$ ($\kappa = \mathrm{x}, \mathrm{s}, \mathrm{p}$) has been substituted in (21-23) with $g^*_{\kappa,l} = W_\mathrm{r}(\frac{1}{2}\omega_{\kappa,l}/\pi)\,g_{\kappa,l}$, where

$$W_\mathrm{r}(f) = \begin{cases} 1 & : \ f < f_\mathrm{r} \\ (f_\mathrm{n} - f)/(f_\mathrm{n} - f_\mathrm{r}) & : \ f_\mathrm{r} \le f < f_\mathrm{n} \\ 0 & : \ f \ge f_\mathrm{n} \end{cases} \tag{28}$$

is a frequency window in which $f_\mathrm{n} = \frac{1}{2}\Delta_t^{-1}$ is the Nyquist frequency and $f_\mathrm{r} < f_\mathrm{n}$ is the highest mode frequency to be rendered in full amplitude. This allows variation of system parameters over time without mode aliasing and - control rate permitting - without causing significant discontinuities in the output sigal. For audio rates of 44.1 kHz or higher, a sensible choice is to set $f_\mathrm{r} = 20$ kHz.

3.2. A Vector-Matrix Update Form

Stacking all modal states of the string and plate in column vectors ($\bar{\mathbf{u}}_\mathrm{s}^n$, $\bar{\mathbf{q}}_\mathrm{s}^n$) and ($\bar{\mathbf{u}}_\mathrm{p}^n$, $\bar{\mathbf{q}}_\mathrm{p}^n$), the system modal state vectors can be defined as

$$\bar{\mathbf{u}}^n = [(\bar{\mathbf{u}}_\mathrm{s}^n)^\mathsf{T},\, u_\mathrm{b},\, (\bar{\mathbf{u}}_\mathrm{p}^n)^\mathsf{T}]^\mathsf{T}, \quad \bar{\mathbf{q}}^n = [(\bar{\mathbf{q}}_\mathrm{s}^n)^\mathsf{T},\, q_\mathrm{b},\, (\bar{\mathbf{q}}_\mathrm{p}^n)^\mathsf{T}]^\mathsf{T}, \tag{29}$$

where $\bar{q}_{\kappa,l} = (\Delta_t/(2m_\kappa))\bar{p}_{\kappa,l}$ is a convenient change of variable. The complete discrete-time system can then be written

$$\delta \bar{\mathbf{u}} = \mu \bar{\mathbf{q}}, \tag{30}$$

$$\delta \bar{\mathbf{q}} = -\left(\mathbf{A}\mu + \mathbf{B}\delta\right)\bar{\mathbf{u}} + \mathbf{\Xi}\left(\mathbf{G}_\mathrm{c}\mathbf{F}^{n+\frac{1}{2}} + \mathbf{G}_\mathrm{e}\mathbf{F}_\mathrm{e}^{n+\frac{1}{2}} + \mathbf{G}_\mathrm{d}F_\mathrm{d}^{n+\frac{1}{2}}\right), \tag{31}$$

where $\mathbf{F}^{n+\frac{1}{2}} = [F_1^{n+\frac{1}{2}}\ F_2^{n+\frac{1}{2}}]^\mathsf{T}$ and $\mathbf{F}_\mathrm{e}^{n+\frac{1}{2}} = [F_\mathrm{x}^{n+\frac{1}{2}}\ F_\mathrm{b}^{n+\frac{1}{2}}]^\mathsf{T}$ are force vectors and where the matrices

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_\mathrm{s} & \mathbf{0}_\mathrm{s} & \mathbf{0}_\mathrm{sp} \\ 0 & 0 & 0 \\ \mathbf{0}_\mathrm{ps} & \mathbf{0}_\mathrm{p} & \mathbf{A}_\mathrm{p} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \mathbf{B}_\mathrm{s} & \mathbf{0}_\mathrm{s} & \mathbf{0}_\mathrm{sp} \\ 0 & b_\mathrm{b} & 0 \\ \mathbf{0}_\mathrm{ps} & \mathbf{0}_\mathrm{p} & \mathbf{B}_\mathrm{p} \end{bmatrix}, \tag{32}$$

contain submatrices $\mathbf{A}_\kappa$ and $\mathbf{B}_\kappa$ that are $M_\kappa \times M_\kappa$ diagonal matrices with diagonal elements $A_{\kappa,l,l} = a^*_{\kappa,l}$ and $B_{\kappa,l,l} = b^*_{\kappa,l}$,

respectively, and with $b_\mathrm{b} = (r_\mathrm{b}\Delta_t)/(2m_\mathrm{b})$. The other matrices in (31) are

$$\mathbf{G}_\mathrm{c} = \begin{bmatrix} \mathbf{g}_\mathrm{s} & \mathbf{0}_\mathrm{s} \\ -1 & 1 \\ \mathbf{0}_\mathrm{p} & -\mathbf{g}_\mathrm{p} \end{bmatrix}, \quad \mathbf{G}_\mathrm{e} = \begin{bmatrix} \mathbf{g}_\mathrm{x} & \mathbf{0}_\mathrm{s} \\ 0 & 1 \\ \mathbf{0}_\mathrm{p} & \mathbf{0}_\mathrm{p} \end{bmatrix},$$
$$\mathbf{G}_\mathrm{d} = \begin{bmatrix} \mathbf{g}_\mathrm{d} \\ 0 \\ \mathbf{0}_\mathrm{p} \end{bmatrix}, \quad \boldsymbol{\Xi} = \begin{bmatrix} \xi_\mathrm{s}\mathbf{I}_\mathrm{s} & \mathbf{0}_\mathrm{s} & \mathbf{0}_\mathrm{sp} \\ 0 & \xi_\mathrm{b} & 0 \\ \mathbf{0}_\mathrm{ps} & \mathbf{0}_\mathrm{p} & \xi_\mathrm{p}\mathbf{I}_\mathrm{p} \end{bmatrix}, \tag{33}$$

where matrices $\mathbf{G}_{\mathrm{c,e,d}}$ feature modal column vectors $\mathbf{g}_\kappa$ which have M_κ elements defined by (15), $\xi_\kappa = \Delta_t^2/(2m_\kappa)$, and where $\mathbf{I}_\kappa$ are $M_\kappa \times M_\kappa$ identity matrices. By setting $\bar{\mathbf{s}} = \delta\bar{\mathbf{u}} = \mu\bar{\mathbf{q}}$ from (30), equation (31) can be reworked into

$$\bar{\mathbf{s}} = \bar{\mathbf{e}} + \mathbf{H}_\mathrm{c}\mathbf{F}^{n+\frac{1}{2}} + \mathbf{H}_\mathrm{d}F_\mathrm{d}^{n+\frac{1}{2}}, \tag{34}$$

where

$$\bar{\mathbf{e}} = 2\mathbf{C}\left(\bar{\mathbf{q}}^n - \mathbf{A}\bar{\mathbf{u}}^n\right) + \mathbf{H}_\mathrm{e}\mathbf{F}_\mathrm{e}^{n+\frac{1}{2}}, \tag{35}$$

and

$$\mathbf{H}_\mathrm{c} = \begin{bmatrix} \mathbf{h}_\mathrm{s} & \mathbf{0}_\mathrm{s} \\ -\xi_\mathrm{b} & \xi_\mathrm{b} \\ \mathbf{0}_\mathrm{p} & -\mathbf{h}_\mathrm{p} \end{bmatrix}, \mathbf{H}_\mathrm{e} = \begin{bmatrix} \mathbf{h}_\mathrm{x} & \mathbf{0}_\mathrm{s} \\ 0 & \xi_\mathrm{b} \\ \mathbf{0}_\mathrm{p} & \mathbf{0}_\mathrm{p} \end{bmatrix},$$
$$\mathbf{H}_\mathrm{d} = \begin{bmatrix} \mathbf{h}_\mathrm{d} \\ 0 \\ \mathbf{0}_\mathrm{p} \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} \mathbf{C}_\mathrm{s} & \mathbf{0}_\mathrm{s} & \mathbf{0}_\mathrm{sp} \\ 0 & (1+b_\mathrm{b})^{-1} & 0 \\ \mathbf{0}_\mathrm{ps} & \mathbf{0}_\mathrm{p} & \mathbf{C}_\mathrm{p} \end{bmatrix}, \tag{36}$$

with, for $\kappa = \mathrm{s}, \mathrm{p}$, we have $\mathbf{C}_\kappa = \left(\mathbf{I}_\kappa + \mathbf{A}_\kappa + \mathbf{B}_\kappa\right)^{-1}$, and

$$\mathbf{h}_\mathrm{s} = \xi_\mathrm{s}\mathbf{C}_\mathrm{s}\mathbf{g}_\mathrm{s}, \quad \mathbf{h}_\mathrm{p} = \xi_\mathrm{p}\mathbf{C}_\mathrm{p}\mathbf{g}_\mathrm{p}, \quad \mathbf{h}_\mathrm{d} = \xi_\mathrm{s}\mathbf{C}_\mathrm{s}\mathbf{g}_\mathrm{d}, \quad \mathbf{h}_\mathrm{x} = \xi_\mathrm{s}\mathbf{C}_\mathrm{s}\mathbf{g}_\mathrm{x}, \tag{37}$$

Once known, the step vector $\bar{\mathbf{s}}$ is employed to update the modal states with

$$\bar{\mathbf{u}}^{n+1} = \bar{\mathbf{s}} + \bar{\mathbf{u}}^n, \quad \bar{\mathbf{q}}^{n+1} = \bar{\mathbf{s}} - \bar{\mathbf{q}}^n, \tag{38}$$

after which the audio output signal vector can be computed as

$$\mathbf{p}_\mathrm{a}^n = \mathbf{W}_\mathrm{a}\mathbf{q}_\mathrm{p}^n, \tag{39}$$

where $\mathbf{W}_\mathrm{a} = (2m_\mathrm{p})\Delta_t^{-1}\left[\mathbf{g}_{\mathrm{a},1}, \mathbf{g}_{\mathrm{a},2}, \cdots \mathbf{g}_{\mathrm{a},K}\right]^\mathsf{T}$ and where $\mathbf{g}_{\mathrm{a},k}$ is the plate modal column vector for the pickup position $(x_{\mathrm{a},k}, y_{\mathrm{a},k})$. Note that all of the $M \times M$ matrices ($\mathbf{A}$,$\mathbf{B}$, $\mathbf{C}$ and $\boldsymbol{\Xi}$) featured above , with $M = M_\mathrm{s} + M_\mathrm{p} + 1$, are diagonal, and as such can be replaced with $M \times 1$ vectors, with the calculation in (35) being achieved using elementwise vector multiplication.

3.3. Solving for the Step Vector

To find a way to compute the step vector $\bar{\mathbf{s}}$, we first eliminate the damper force from the system. From (24), one may write

$$F_\mathrm{d}^{n+\frac{1}{2}} = -r_\mathrm{d}\Delta_t^{-1}s_\mathrm{d}, \tag{40}$$

where $s_\mathrm{d} = \delta u_\mathrm{d}$, which (from multiplying (34) with $\mathbf{G}_\mathrm{d}^\mathsf{T}$) also equals

$$s_\mathrm{d} = \mathbf{G}_\mathrm{d}^\mathsf{T}\left(\bar{\mathbf{e}} + \mathbf{H}_\mathrm{c}\mathbf{F}^{n+\frac{1}{2}}\right) + \mathbf{G}_\mathrm{d}^\mathsf{T}\mathbf{H}_\mathrm{d}F_\mathrm{d}^{n+\frac{1}{2}}. \tag{41}$$

Eliminating both s_d and $F_\mathrm{d}^{n+\frac{1}{2}}$ then allows re-writing (34) as

$$\bar{\mathbf{s}} = \bar{\mathbf{w}} + \boldsymbol{\Phi}_\mathrm{c}\mathbf{F}^{n+\frac{1}{2}}, \tag{42}$$

where $\bar{\mathbf{w}} = \bar{\mathbf{e}} - \theta_\mathrm{d}\mathbf{H}_\mathrm{d}\mathbf{G}_\mathrm{d}^\mathsf{T}\bar{\mathbf{e}}$ and $\boldsymbol{\Phi}_\mathrm{c} = \mathbf{H}_\mathrm{c} - \theta_\mathrm{d}\mathbf{H}_\mathrm{d}\mathbf{G}_\mathrm{d}^\mathsf{T}\mathbf{H}_\mathrm{c}$, with $\theta_\mathrm{d} = r_\mathrm{d}/(r_\mathrm{d}\mathbf{G}_\mathrm{d}^\mathsf{T}\mathbf{H}_\mathrm{d} + \Delta_t)$. Next, the modal coordinate equation (42) is transformed into an equation in the spring connection coordinates by pre-multiplying (42) with $-\mathbf{G}_\mathrm{c}^\mathsf{T}$, which gives

$$\boldsymbol{\mathcal{M}}\mathbf{F}^{n+\frac{1}{2}} = \mathbf{w} - \mathbf{s}, \tag{43}$$

where $\mathbf{w} = -\mathbf{G}_\mathrm{c}^\mathsf{T}\bar{\mathbf{w}}$, $\mathbf{s} = \delta\mathbf{u}$ (with $\mathbf{u}^n = [u_1^n,\, u_2^n]^\mathsf{T}$), and

$$\boldsymbol{\mathcal{M}} = \mathbf{G}_\mathrm{c}^\mathsf{T}\boldsymbol{\Phi}_\mathrm{c} = \begin{bmatrix} (\xi_\mathrm{b} + \phi_\mathrm{s}) & -\xi_\mathrm{b} \\ -\xi_\mathrm{b} & (\xi_\mathrm{b} + \phi_\mathrm{p}) \end{bmatrix}, \tag{44}$$

with $\phi_\mathrm{s} = \mathbf{g}_\mathrm{s}^\mathsf{T}\mathbf{h}_\mathrm{s} - \theta_\mathrm{d}\mathbf{g}_\mathrm{s}^\mathsf{T}\mathbf{h}_\mathrm{d}\mathbf{g}_\mathrm{d}^\mathsf{T}\mathbf{h}_\mathrm{s}$ and $\phi_\mathrm{p} = \mathbf{g}_\mathrm{p}^\mathsf{T}\mathbf{h}_\mathrm{p}$. From the second equation in (24), one may write

$$\mathbf{F}^{n+\frac{1}{2}} = \boldsymbol{\lambda}(\mathbf{s}) + k_\mathrm{L}\left(\frac{1}{2}\mathbf{s} + \mathbf{u}^n\right), \tag{45}$$

where $\boldsymbol{\lambda}(\mathbf{s}) = [\lambda_1(s_1),\, \lambda_2(s_2)]^\mathsf{T}$ with, for $\ell = 1, 2$

$$\lambda_\ell(s_\ell) = \frac{V_\ell(s_\ell + u_\ell^n) - V_\ell(u_\ell^n)}{s_\ell}. \tag{46}$$

After substituting (45) into (43), a system of two coupled nonlinear simultaneous equations is obtained:

$$\boldsymbol{\mathcal{M}}\boldsymbol{\lambda}(\mathbf{s}) + \left(\mathbf{I} + \frac{1}{2}k_\mathrm{L}\boldsymbol{\mathcal{M}}\right)\mathbf{s} + \boldsymbol{\iota} = \mathbf{0}, \tag{47}$$

where $\boldsymbol{\iota} = k_\mathrm{L}\boldsymbol{\mathcal{M}}\mathbf{u}^n - \mathbf{w}$. These can be solved iteratively using Newton's method. Once $\mathbf{s}$ is solved, the spring forces are updated from (43), after which the step vector $\bar{\mathbf{s}}$ is calculated with (42). Besides updating the states with (38) and the output with (39), it is also required that the connection displacement vector is updated as $\mathbf{u}^{n+1} = \mathbf{s} + \mathbf{u}^n$.

3.4. Uniqueness and Convergence

Defining $\lambda'_\ell = \frac{\partial\lambda_\ell}{\partial s_\ell} \geq 0$, the Jacobian to be used for solving (47) is

$$\mathbf{J} = \begin{bmatrix} 1 + \left(\xi_\mathrm{b} + \phi_\mathrm{s}\right)\left(\lambda'_1 + \frac{k_\mathrm{L}}{2}\right) & -\xi_\mathrm{b}\left(\lambda'_2 + \frac{k_\mathrm{L}}{2}\right) \\ -\xi_\mathrm{b}\lambda'_1 + \left(\frac{k_\mathrm{L}}{2}\right) & 1 + \left(\xi_\mathrm{b} + \phi_\mathrm{p}\right)\left(\lambda'_2 + \frac{k_\mathrm{L}}{2}\right) \end{bmatrix} \tag{48}$$

which is clearly positive definite for any positive bridge mass (meaning ξ_b is finite), hence (47) has a unique root $\mathbf{s}^*$. $\mathbf{J}$ is also a so-called M-matrix, which is a condition for global convergence [15]. From the fact that $\lambda_\ell(s_\ell)$ has a single inflection point (i.e. concave for $s_\ell < 0$ and convex for $s_\ell > 0$), it then follows that there is guaranteed monotic convergence from any $\mathbf{s}$ for which each of the elements s_ℓ has the same sign as the corresponding element of the actual root and satisfies $|s_\ell| \geq |s_\ell^*|$ (see [15], p. 112-113). Empirically observing that the solver generally reaches this condition from any starting position then suggests global convergence. Such robustness indeed appears to hold, as extensive testing with randomised starting points has indicated that the solver generally converges (no cases of non-convergence observed for the tested parameters sets).

What is, however, not guaranteed without any further conditions is that the iterative solver converges within a fixed number of iterations. The iteration count in fact depends not only on the initial guess, but also on the driving signal amplitude and all system parameters. A practical way to try and keep the iteration count under control is by empirically setting constraints on the parameters (see Table 1) and applying a limiter on the input signal.

3.5. Stability

The system energy at time instant $t = n\Delta_t$ is the sum of the modal energies of the string and plate plus the kinetic energy of the bridge mass and the potential energies in the linear and nonlinear spring elements:

$$H^n = \sum_{l=1}^{M_\mathrm{s}} \left[\frac{(\bar{p}^n_{\mathrm{s},l})^2}{2m_\mathrm{s}} + \frac{k^*_{\mathrm{s},l}(\bar{u}^n_{\mathrm{s},l})^2}{2} \right] + \sum_{l=1}^{M_\mathrm{p}} \left[\frac{(\bar{p}^n_{\mathrm{s},l})^2}{2m_\mathrm{p}} + \frac{k^*_{\mathrm{p},l}(\bar{u}^n_{\mathrm{s},l})^2}{2} \right] + \frac{(\bar{p}^n_\mathrm{b})^2}{2m_\mathrm{b}} + \frac{k_\mathrm{L}}{2}\left(u_1^2 + u_2^2\right) + V_1(u_1^n) + V_2(u_2^n). \quad (49)$$

Given that all the individual terms in (49) are non-negative, it follows that $H^n \geq 0$. In vector-matrix form, H^n can be written

$$H^n = (\bar{\mathbf{q}}^n)^\mathsf{T} \boldsymbol{\Xi}^{-1} \bar{\mathbf{q}}^n + (\bar{\mathbf{u}}^n)^\mathsf{T} \boldsymbol{\Xi}^{-1} \mathbf{A} \bar{\mathbf{u}}^n + \frac{k_\mathrm{L}}{2} \|\mathbf{u}^n\|_2 + \|\mathbf{V}(\mathbf{u}^n)\|_1. \quad (50)$$

A numerical power balance that holds under time-invariant parameters is obtained by first pre-multiplying (31) with $\boldsymbol{\Xi}^{-1}$ and, subsquently, the left-hand side with $(\mu\mathbf{q})^\mathsf{T}$ and the right-hand side by $(\delta\mathbf{u})^\mathsf{T}$, which yields

$$\frac{\delta H}{\Delta_t} = \frac{H^{n+1} - H^n}{\Delta_t} = P^{n+\frac{1}{2}} - Q^{n+\frac{1}{2}}, \quad (51)$$

where $P^{n+\frac{1}{2}} = \frac{1}{2}\Delta_t^{-1} \mathbf{G}_\mathrm{e}^\mathsf{T}\, \delta\bar{\mathbf{u}}\, \mathbf{F}_\mathrm{e}^{n+\frac{1}{2}}$ is the input power and

$$Q^{n+\frac{1}{2}} = \frac{(\delta\bar{\mathbf{u}})^\mathsf{T} \boldsymbol{\Xi}^{-1} \mathbf{B} \delta\bar{\mathbf{u}}}{\Delta_t} + r_\mathrm{b} \frac{(\delta\mathbf{u})^\mathsf{T}\delta\mathbf{u}}{\Delta_t^2} + r_\mathrm{d} \frac{(\delta u_\mathrm{d})^2}{\Delta_t^2} \geq 0 \quad (52)$$

is the dissipated power. Hence in the absence of input power, the system energy cannot grow. The fact that all terms in (50) are non-negative and either quadratic or monotone functions of one of the variables implies bounds on u^n and q^n, confirming numerical stability with no further conditions on the temporal step Δ_t. The above power balance does not hold under time variation of the system parameters. Hence under continuous parameter update, energy growth can potentially occur if the power injected by such variation outstrips the dissipation; our observations in off-line experiments are that this occurs only when parameters are varied at rates much higher than required for parametric control. A more thorny issue potentially arises when, during real-time operation, the iteration count spikes due to fast parameter sweeps. This may lead to (47) not being solved to high accuracy for a number of time instants, which in turn can cause violations of the power balance. The associated instability risk is currently managed in ad-hoc fashion by empirically constraining the rate at which parameters are varied (employing low-pass filtering at control rate of all the parameter signals).

4. PARAMETER SPACE EXPLORATION

4.1. Control Parameters

Exposing the user to the full set of parameters featuring in (1-3) makes it unnecessarily difficult to learn navigating the parameter space because of parameter redundancy. Without loss of generality, the parameter set is therefore reduced here 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

Table 1: *Tunable parameters, the constraints imposed upon them for real-time operation, and example values.*

		Fig. 4	Fig. 5	Fig. 6
STRING				
fundam. freq. [Hz]	$0 < \tilde{f}_\mathrm{s} < \frac{\pi}{\Delta_t}$	100	47.3	80
inharmonicity coeff.	$0 \leq \mathcal{B}_\mathrm{s}$	10^{-5}	10^{-5}	10^{-5}
damping [s^{-1}]	$0 \leq \sigma_{\mathrm{s},0}$	1	2	0.5
damping [m/s]	$0 \leq \sigma_{\mathrm{s},1}$	10^{-3}	$4{\cdot}10^{-4}$	10^{-2}
damping [m^3/s]	$0 \leq \sigma_{\mathrm{s},3}$	10^{-5}	$4{\cdot}10^{-6}$	10^{-4}
damper [s^{-1}]	$0 \leq \sigma_\mathrm{d}$	500	0	0
connection position	$0 < x'_\mathrm{c} < 1$	0.87	0.93	0.98
damper position	$0 < x'_\mathrm{d} < 1$	0.99	0.99	0.99
excitation position	$0 < x'_\mathrm{x} < 1$	0.07	0.5	0.5
BRIDGE				
modal mass ratio	$10^{-4} < R_\mathrm{bs}$	6|0.6	10^{-4}	1
damping [s^{-1}]	$0 \leq \zeta_\mathrm{b}$	1	2	10^{-2}
stiffness	$0 \leq k_\mathrm{b} \leq 10^6$	10^5	10^5	10^6
nonlinearity	$0 \leq \eta \leq 1$	0	0|1	1
nonlinear exponent	$1 \leq \alpha \leq 3$	1	3	1.1
gravity [m/s^2]	$\lvert g_\mathrm{b}\rvert \leq 10$	0	0	-0.5
spring push level	$0 \leq G_1^+ \leq 1$	1	1	1
spring pull level	$0 \leq G_1^- \leq 1$	1	1	0
spring push level	$0 \leq G_2^+ \leq 1$	1	1	1
spring pull level	$0 \leq G_2^- \leq 1$	1	1	0
PLATE				
fundam. freq. [Hz]	$0 < \tilde{f}_\mathrm{p} < \frac{\pi}{\Delta_t}$	17.7	50	30
dimensional ratio	$0 < R_{xy}$	0.89	0.98	0.77
modal mass ratio	$0 < R_\mathrm{ps}$	10	1	10
damping [s^{-1}]	$0 \leq \sigma_{\mathrm{p},0}$	20	2	4
damping [m/s]	$0 \leq \sigma_{\mathrm{p},1}$	10^{-4}	$4{\cdot}10^{-4}$	10^{-2}
damping [m^3/s]	$0 \leq \sigma_{\mathrm{p},3}$	10^{-6}	$4{\cdot}10^{-6}$	10^{-4}
connection position	$0 < x'_\mathrm{c} < 1$	0.61	0.17	0.61
connection position	$0 < y'_\mathrm{c} < 1$	0.50	0.11	0.43
pickup position	$0 < x'_{\mathrm{a},k} < 1$	0.13	0.13	0.13
pickup position	$0 < y'_{\mathrm{a},k} < 1$	0.93	0.93	0.93

are then introduced with the aim of enabling intuitive control of the string, bridge, and plate characteristics:

$$\tilde{f}_\mathrm{s} = f_{\mathrm{s},1} = \frac{1}{2}\sqrt{\left(\frac{E_\mathrm{s} I_\mathrm{s}}{\rho_\mathrm{s} A_\mathrm{s}}\right)\pi^2 + \left(\frac{T_\mathrm{s}}{\rho_\mathrm{s} A_\mathrm{s}}\right)}, \quad (53)$$

$$\mathcal{B}_\mathrm{s} = \pi^2 \frac{E_\mathrm{s} I_\mathrm{s}}{T_\mathrm{s}}, \quad \zeta_\mathrm{d} = \frac{r_\mathrm{d}}{2m_\mathrm{s}}, \quad z'_\kappa = z_\kappa \ (\kappa = \mathrm{x}, \mathrm{c}, \mathrm{d}), \quad (54)$$

$$R_\mathrm{bs} = \frac{m_\mathrm{b}}{m_\mathrm{s}}, \quad \zeta_\mathrm{b} = \frac{r_\mathrm{b}}{2m_\mathrm{b}}, \quad R_\mathrm{ps} = \frac{m_\mathrm{p}}{m_\mathrm{s}}, \quad R_{xy} = \frac{L_x}{L_y}, \quad (55)$$

$$\tilde{f}_\mathrm{p} = f_{\mathrm{p},1,1} = \frac{1}{2}\sqrt{\frac{G_\mathrm{p}\pi^2}{\rho_\mathrm{p} h_\mathrm{p}}}\left(L_x^{-2} + L_y^{-2}\right), \quad (56)$$

$$x'_\mathrm{c} = \frac{x_\mathrm{c}}{L_x}, \quad y'_\mathrm{c} = \frac{y_\mathrm{c}}{L_y}, \quad x'_{\mathrm{a},k} = \frac{x_{\mathrm{a},k}}{L_x}, \quad y'_{\mathrm{a},k} = \frac{y_{\mathrm{a},k}}{L_y}. \quad (57)$$

Furthermore, the connection spring stiffness constants in (4) are parameterised as follows:

$$k_\mathrm{L} = (1-\eta)k_\mathrm{b}, \quad k_\ell^\pm = \eta\, k_\mathrm{b} G_\ell^\pm \cdot 10^{4(\alpha-1)}, \quad (58)$$

where k_b is an overall bridge stiffness parameter, $0 \leq G_\ell^\pm \leq 1$ set the relative push and pulling levels of the spring and $0 \leq \eta \leq 1$ gives control over the level of nonlinear behaviour. The formulation in (58) helps ensuring, in combination with the constraints imposed on the parameters, that the overall stiffness does not exceed

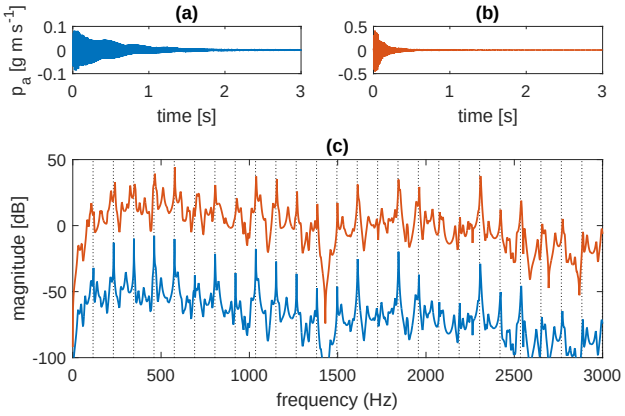

Figure 4: *Example of linear coupling between the string, bridge and plate (see Table 1 for the model parameters). (a): large bridge mass ($R_{\mathrm{bs}} = 6$). (b): small bridge mass ($R_{\mathrm{bs}} = 0.6$). (c): corresponding magnitude spectra. The upper curve ($R_{\mathrm{bs}} = 0.6$) is offset by 50dB for clarity. The vertical dotted lines indicate the positions of the string mode frequencies.*

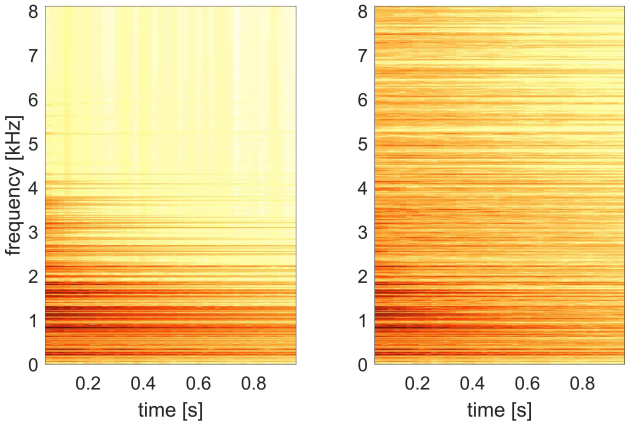

Figure 5: *Example of plate-like sounds via strong string-plate coupling (see Table 1 for the model parameters). Left: plate momentum spectrogram for $\eta = 0$. Right: plate momentum spectrogram for $\eta = 1$.*

certain levels which would cause an excessive amount of iterations to be completed by the iterative solver. Finally, the external bridge force is recast here as a gravitational force $F_{\mathrm{b}}(t) = g_{\mathrm{b}} m_{\mathrm{b}}$, where g_{b} is a gravitational acceleration value that can be varied at control rate by the user. A full list of tunable parameters is shown in Table 1, including the applied value constraints.

4.2. Numerical Experiments

The simplest configuration of the bridge is to use strictly linear spring connections (i.e. $\eta = 0$). The plots in Fig. 4 show signals and magnitude spectra of such a case when driving the string with a short smooth pulse. With a relative large bridge mass (setting $R_{\mathrm{bs}} = 6$), strong standing waves develop in the string, and the transfer of energy is largely uni-directional, i.e. from the string to the plate, much in the way conventional string instruments operate. As can be seen in the corresponding spectrum (lower curve in Fig. 4(c)), the output also exhibits plate modes, which are more damped than the string resonances. Setting the bridge mass to a smaller value ($R_{\mathrm{bs}} = 0.6$) leads to increased coupling between the plate and the string, and the system energy is then dissipated more quickly (see Fig. 4(b)), yielding an output in which the string modes are less dominant over the plate modes (see upper curve of Fig. 4(c)).

The notion of effecting strong string-plate coupling can be taken to its extreme by setting R_{bs} to a negligibly small value and R_{ps} to unity, while using similar damping values for the plate and string. The coupling is now very much bi-directional, and any standing waves on the string no longer correspond to the string eigenmodes. Although different than that of the modelled plate, the distribution of system modes across the frequency axis is nevertheless similar to that of a plate, so the perceived sound output is plate-like. Hence in this configuration, the string is effectively merely a mechanical interface to a plate-like instrument. An interesting feature to explore with this type of connection is to introduce nonlinear behaviour by varying the η parameter. Fig. 5 shows the spectrograms resulting for $\eta = 0$ (linear springs) and $\eta = 1$

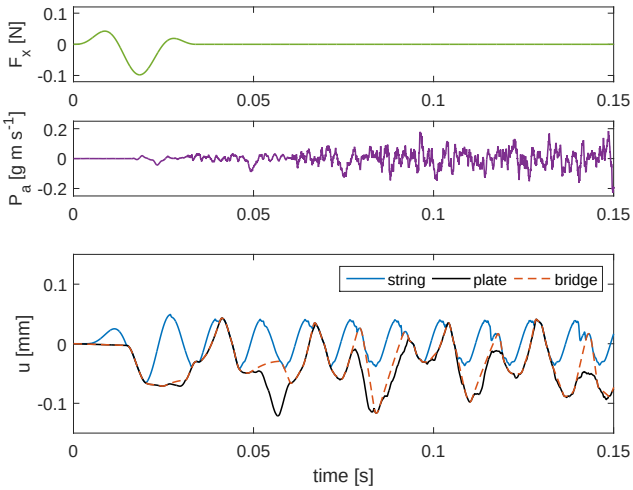

Figure 6: *Example of a rattling bridge (see Table 1 for the model parameters). Top: string excitation force signal. Middle: plate momentum monitored at the pick-up position. Bottom: displacement at the connection point of the string, plate, and bridge mass.*

(fully nonlinear springs), with $\alpha = 3$. As can be seen, the second case exhibits quick growth of densely spaced high-frequency partials, indicating strong nonlinear inter-mode coupling, and yielding a somewhat cymbal-like sound output.

An altogether different type of nonlinear behaviour is explored when imparting asymmetries in either or both of the spring connections, which introduces rattling effects. For example, setting $G_1^- = G_2^- = 0$ and $G_1^+ = G_2^+ = 1$ amounts to a bridge that is in contact with the plate and the string at equilibrium but that - in the absence of forces that pull it back to either object when it moves away from them - is free to rattle during vibration. Special effects are obtained when letting the string vibrate at a low frequency, as such enabling the generation of slowly evolving rattling patterns which partly play out at sub-sonic rates. An example case is demonstrated in Fig. 6, with the lower plot showing the complex motion of the bridge mass and its impactive interaction

```
// initialise output value with zero
pa = 0;
// loop over plate modes, index starts at Ms+1
for (int i = 0; i < Mp; i++) {
   pa += *(wa + i) * *(q_modes + Ms + 1 + i);
   }
```

Figure 7: *C++ code for implementing (39) for a single output.*

```
// initialise __mm256 block with zeros
block = _mm256_set_pd(0, 0, 0, 0);
// loop over plate modes, sum blocks of four
for (int i = 0; i < Mp; i += 4) {
   block = _mm256_fmadd_pd(_mm256_load_pd(&wa[i]),
        _mm256_load_pd(&q_modes[Ms+1+i]), block);
}
// store final block results and sum over them
temp = _mm256_store_pd(block);
pa = temp[0] + temp[1] + temp[2] + temp[3];
```

Figure 8: *AVX code for implementing (39) for a single output.*

with the string and plate, in response to a short windowed 40 Hz sine wave. For the chosen parameters (including gravity), the resulting sound is somewhat reminiscent of a snare drum roll. More generally, this type of configuration produces rattling and buzzing patterns of a semi-chaotic nature. Sound examples of all three of the above cases are available on the companion website[1] alongside various further explorative sounds and supporting material.

4.3. Real-Time Implementation

For real-time rendering, the system was built in Audio Unit plug-in architecture, coding in C++ within the JUCE framework [16]. JUCE yields executable code within a standard plug-in API provided by Core Audio, and supports a host of plugin formats. In order to increase the number of modes that can be run in real-time, a second version of the code uses Advanced Vector Extensions (AVX) for the parts of the code that loop over M_s string modes or M_p plate modes. This allows performing arithmetic operations over multiple variables simultaneously; for double precision on standard processors, most of which currently use 256 bit registers, this means the code can operate on four modes at a time. To exemplify the coding difference, Figs. 7 and 8 show the instructions used in C++ and in AVX, respectively, for implementing the matrix operation in (39) when running the system with a single audio output. The iterative solver, which does not permit similar parallelisation, takes upto about 5% of the computation when the exit condition chosen as $||\mathbf{s}_{j+1} - \mathbf{s}_j|| < 10^{-15}$, where j indicates the iteration.

The run-time computations also include re-calculation of system coefficients, such as the non-zero elements of $\mathbf{A}$, $\mathbf{C}$, $\mathbf{h}_\mathrm{c}$, and $\mathbf{h}_\mathrm{x}$, according to parameter changes made by the user at regular intervals. The shorter the audio buffer size, the faster the parameters can be varied by the user without significant artefacts, but the more this adds to the overall load. Table 2 lists the results of testing the number of modes that each code version can render in real-time without underflow when fixing the ratio between the number of string and plate modes as $M_\mathrm{p} \approx 4M_\mathrm{s}$, for different buffer sizes. Both codes were compiled with optimisation level -O3 and using AVX2. The first row shows the maximum number of modes when not performing any parameter updates during runtime. Similar to

[1] `http://www.socasites.qub.ac.uk/mvanwalstijn/dafx17a/`

Table 2: *Maximum number of system modes ($M = M_\mathrm{p} + M_\mathrm{s} + 1$) for the plugin running at 44.1 kHz when choosing $M_\mathrm{p} \approx 4M_\mathrm{s}$.*

audio buffer length (samples)	**C++**	**C++ with AVX**
no updates	3207	7037
512	2930	6505
256	2637	5925
128	2476	5253
64	2171	4571

the findings by Webb and Bilbao [8] for a finite difference based system, the AVX instructions accelerate the computations by about a factor two for double-precision floating point. The viability of single-precision AVX acceleration (which can be expected to provide a 4 times speed up, but may introduce round-off noise) is currently being investigated. Note that for both the C++ and the C++ with AVX code, the parameter updates were written entirely in C++. Further investigation could establish whether this part of the code can also be optimised by replacing C++ code with AVX instructions.

A relatively large number of string modes is chosen here because this enables interesting coupling phenomena. Experiments have indicated that a total of 5000 system modes is sufficient to generate a musically appropriate range of plate and string sizes (indeed, each of the sound examples available on the companion webpage were generated with this number of system modes). For a 44.1 kHz sample rate, this means that the mode series of larger plates is truncated before Nyquist, but - as reported in [11] - the resulting loss of high-frequency detail is not necessarily perceptually significant.

4.4. Control Interface

To facilitate ergonomic real-time control, each of the parameters is knob-controlled using a Knobbee board [17], which supports OSC messages at 10 bit resolution. Besides navigating the parameter space in search of interesting sounds, the user can employ the board to make gestures via parameter changes, many of which are not possible or difficult to achieve with a real-world counterpart of the model. The string can be excited either using a pre-stored signal or more interactively, using a silent string controller object based on the design proposed in [18]. Elsewhere [19] we report preliminary findings regarding the artistic use of these control interfaces with an earlier, simpler, bridge-less version of the model.

5. CONCLUSIONS AND PERSPECTIVES

The string-bridge-plate model does not enable faithful emulation of any existing musical instrument; instead, it offers synthesis of a range of sounds of an inherently mechano-acoustic nature. The novelty resides mainly in the design of the bridge, which incorporates parametrically detachable nonlinear spring connections on either side of the bridge mass; this is modelled numerically in a way that allows real-time simulation of the complex, semi-chaotic rattling and buzzing that entails when reconfiguring these connections on the fly. The realisation of this design extends on what existing real-time physical model implementations and environments currently offer in terms of contact dynamics. Further versatility 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 inter-object coupling through adjusting the modal mass values.

The tunability of the parameters indirectly relies on the modal expansion approach, which - as explained in § 3.1 - enables eliminating the dispersion and attenutation errors inherent to the unconditionally stable numerical scheme. The modal form has further beneficial features, in that (a) it facilitates direct control of frequency-dependent damping, (b) the algorithmic efficiency is greatly helped by the sparsity of the matrices $\mathbf{G}_\kappa$ and $\mathbf{H}_\kappa$ and the diagonality of matrices $\mathbf{A}$ and $\mathbf{C}$, which is systematically exploited in our implementations of the system, and (c) the number of degrees of freedom (in the present model, the number of modes) can be reduced without affecting the audio bandwidth or introducing numerical dispersion, making the model's computational load easily scalable to the available processing power (though high-frequency detail is lost for larger strings and plates).

There are, however, also limitations to consider. Firstly, a relative large number of system coefficients needs to be regularly updated (compared to, e.g., finite difference schemes). Secondly, the parameter update remains simple and efficient only for objects for which the modal expansion is available in closed-form (e.g. a cantilever beam or a circular membrane), hence extending to more complex and varied boundary conditions is not straightforward. Thirdly, inputs, outputs, and connections are relatively expensive within a modal framework, due to the dot products required to translate between modal and spatial coordinates. The latter is of particular relevance if the generalised form of eqs. (30, 31) were to be taken as the basis for a modular environment.

It is also worthwhile briefly reflecting on the use of a Newton solver in simulating a system featuring multiple nonlinearities, which can present significant challenges, especially when non-smooth forces are involved. As discussed in § 3.4, the form of the nonlinear equation arising in the proposed model has the benefit of global convergence to a unique solution, while placing constraints on the system parameters provides an empirical handle on the iteration count; a useful addition would be to establish a theoretical iteration bound. Altogether this approach goes a long way towards ensuring computational robustness, which is paramount in a real-time synthesis context. Even better robustness would require either an explicit scheme, which - as far as the authors are aware - does not exist in provably stable form for the problem at hand, or an analytically solvable implicit form. The latter does not seem forthcoming either, but does exist for certain simplifications of (4). Further variations, possibly in combination with different discretisation choices, may offer different trade-offs than the ones made in the present study (e.g. sacrificing full tunablity for higher efficiency), and as such are very much worth exploring.

The properties of the string-bridge-plate model are well aligned with design and control tasks, both of which can be performed through navigation of a continuous parameter space in real-time. As such, it offers new opportunities for creative application and control, with potential use in music performance and live improvisation. Hence a natural way forward with research on this topic is to investigate possible extensions, improvements, and variations of the model and its implementation in tandem with developing (and experimenting with) dedicated control interfaces, conducted in close collaboration with performers. The first live performance featuring the string-bridge-plate instrument took place at NIME 2017 [20].

6. REFERENCES

[1] J. O. Smith, "Virtual acoustic musical instruments: Review and update," *Journal of New Music Research*, vol. 33, no. 3, pp. 283–304, 2004.

[2] C. Cadoz, A. Luciani, and J. L. Florens, "CORDIS-ANIMA: A Modeling and Simulation System for Sound and Image Synthesis: The General Formalism," *Computer Music Journal*, vol. 17, no. 1, pp. 19–29, 1993.

[3] J. Derek Morrison and J. M. Adrien, "MOSAIC: A Framework for Modal Synthesis," *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, 1993.

[4] M. Pearson, "TAO: a physical modelling system and related issues," *Organised Sound*, vol. 1, no. 1, pp. 43–50, 1996.

[5] S. Bilbao, "A Modular Percussion Synthesis Environment," in *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFX-09)*, 2009.

[6] A. S. Allen, *Ruratae: A Physics-Based Audio Engine*, Ph.D. thesis, University of California, San Diego, 2014.

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

[8] C. Webb and S. Bilbao, "On the Limits of Real-Time Physical Modelling Synthesis with a Modular Environment," in *Proc. of the 18th Int. Conf. on Digital Audio Effects (DAFX-15)*, 2015.

[9] S. Bilbao and J. Ffitch, "Prepared Piano Sound Synthesis," in *Proc. of the 9th Int. Conf. on Digital Audio Effects (DAFX-06)*, 2006.

[10] M. Schäfer, P. Frenstátský, and R. Rabenstein, "A Physical String Model with Adjustable Boundary Conditions," in *Proc. of the 19th Int. Conf. on Digital Audio Effects (DAFX-16)*, 2016, pp. 159–166.

[11] S. Orr and M. van Walstijn, "Modal Representation of the Resonant Body within a Finite Difference Framework for Simulation of String Instruments," in *Proc. of the 17th Int. Conf. on Digital Audio Effects (DAFX-09)*, 2009.

[12] M. van Walstijn, J. Bridges, and S. Mehes, "A Real-Time Synthesis Oriented Tanpura Model," in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, 2016, pp. 175–182.

[13] A. Falaize and T. Hélie, "Guaranteed-Passive Simulation of an Electro-Mechanical Piano: A Port-Hamiltonian Approach," in *Proc. of the 18th Int. Conf. on Digital Audio Effects (DAFX-15)*, 2015.

[14] M. van Walstijn and J. Bridges, "Simulation of Distributed Contact in String Instruments: a Modal Expansion Approach," in *Proc. Europ. Sig. Proc Conf (EUSIPCO2016)*, 2016, pp. 1023–1027.

[15] P. Deuflhard, *Newton methods for nonlinear problems: affine invariance and adaptive algorithms*, Springer, 2004.

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

[17] C. Popp and R. Soria-Luz, "Developing Mixer-style Controllers Based On Arduino/Teensy Microcontrollers," in *The 12th Sound and Music Computer Conference*, Maynooth, 2015, pp. 37–41.

[18] S. Mehes, M. van Walstijn, and P. Stapleton, "Towards a Virtual-Acoustic String Instrument," in *13th Sound and Music Computing Conference*, Hamburg, 2016, pp. 286–292.

[19] S. Mehes, M. van Walstijn, and P. Stapleton, "Virtual-Acoustic Instrument Design: Exploring the Parameter Space of a String-Plate Model," in *New Interfaces for Musical Expression*, Copenhagen, 2017, pp. 399–403.

[20] P. Stapleton and A. Pultz Melbye, "VASPI Performance," in *New Interfaces for Musical Expression*, Copenhagen, 2017.