AN EXPLORATIVE STRING-BRIDGE-PLATE MODEL WITH TUNABLE PARAMETERS

Maarten van Walstijn, Sandor Mehes

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 [11]. 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 FAO [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, 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).

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 [3]. An unconditionally stable and modally exact numerical formulation is then developed in [5]. 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

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 [3]. An unconditionally stable and modally exact numerical formulation is then developed in [5]. 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 \( z = z_0 \) along the string axis (see Fig. 1). The string is characterised by its length \( L_s \), mass density \( \rho_s \), cross-sectional area \( A_s \), Young’s modulus \( E_s \), moment of inertia \( I_s \) and damping factor \( \gamma_s \) of yet to be determined wave-number dependency. The plate is of dimensions \( L_p \times L_p \times h_p \), mass density \( \rho_p \), and experiences damping according to \( \gamma_p \). Two of its material properties are encapsulated in the parameter \( G_p = (E_p h_p^3)/(12(1 - \nu_p^2)) \), where \( \nu_p \) is the Poisson ratio. The dynamics of the coupled system are governed by

\[
\rho_s A_s \frac{\partial^2 u_s}{\partial t^2} + T \frac{\partial u_s}{\partial t} - E_s I_s \frac{\partial^4 u_s}{\partial x^4} + \gamma_s \frac{\partial u_s}{\partial t} + \psi_s(z) F_1(t) + \psi_t(z) F_2(t) + \psi_i(z) F_3(t),
\]

\[
\rho_p A_p \frac{\partial^2 u_p}{\partial t^2} = -r_p \frac{\partial u_p}{\partial t} - F_1(t) + F_2(t) + F_3(t),
\]

\[
\rho_p h_p \frac{\partial^2 u_p}{\partial z^2} = -G_p \nabla^4 u_p + \gamma_p \frac{\partial u_p}{\partial z} - \Psi_p(x,y) F_2(t),
\]

where, for \( \kappa = c, x, \alpha, \psi_p(z) = \delta(z - z_a) \) and \( \Delta(x, y) = \delta(x - x_a, y - y_a) \) are single-point spatial distributions, and \( \nabla^4 = \left( \partial^4/\partial x^4 + \partial^4/\partial y^4 \right) \) is a biharmonic operator. It is readily seen that (1) models a beam rather than a string when \( T_s \ll E_s I_s \).

The system is brought into vibration by \( F_3(t) \), which excites the string, and/or \( F_1(t) \) which drives the bridge mass; bridge damping is controlled with \( r_p \). The damper force - which allows suppression of string oscillations on either side of the connection point - is \( F_2(t) = -r_p \frac{\partial u_p}{\partial t} \), where \( u_p(t) = u_p(z_a, t) \). The string connection forces \( F_1 \) and \( F_2 \) are functions of the inter-object distances \( u_s(t) = u_s(z_a, t) \) and \( u_2(t) = u_p(x_a, y_a, t) - u_p(t) \) as follows (\( \ell = 1, 2 \):

\[
F_i(t) = k_{i,1} u_p(t) + k_{i,2} |u_p(t)|^\alpha - k_{i,2}^\alpha |u_2(t)|^\alpha,
\]

where \( |u| = \max(0, u) \), \( k_{i,1} \), \( k_{i,2} \), and \( k_{i,2}^\alpha \) are stiffness coefficients, and \( \alpha \geq 1 \) is a power law exponent. The specific form of (3) 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 \( k_{1,2} \), and \( k_{1,2}^\alpha \), 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 output, we define the plate moment at \( K \) pickup positions (\( x_{a,k}, y_{a,k} \))

\[
p_{a,k} = \rho_p h_p \frac{\partial}{\partial t} u_p(x_{a,k}, y_{a,k}, t).
\]

Figure 2: Examples of spring force-distance curves. (a): linear spring \( [k_s = 10^5, k_p^2 = 0] \); (b): stiffening spring \( [k_s = 6 \times 10^4, k_p^2 = 4 \times 10^4, \alpha = 3] \); (c) one-sided spring \( [k_s = 0, k_p^2 = 1 \times 10^6, \alpha = 1.5] \); (d): asymmetric spring \( [k_s = 0, k_p^2 = 1 \times 10^8, k_p^2 = 2 \times 10^7, \alpha = 1.5] \).

Figure 3: Bridge configuration examples. (a): massless bridge \( [m_b \ll \rho_s A_s L_s, k_s^2 = k_p^2] \); (b): mass-spring bridge \( [k_s^2 = k_p^2] \); (c): flat’ bridge \( [k_s = 0, k_s^2 = k_p^2, k_p^2 > 0, k_p^2 = 0] \); (d): ratting bridge \( [k_s = 0, k_s^2 = k_p^2, k_p^2 > 0, k_s^2 = 0] \).

2.2. Modal Expansion

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

\[
u_s(z, t) = \sum_{i=1}^{M_s} v_{s,i}(z) \bar{u}_{s,i}(t),
\]

\[
u_p(x, y, t) = \sum_{j=1}^{M_p} v_{p,j}(x, y) \bar{u}_{p,j}(t),
\]

where \( v_{s,i} = \sin(\beta_s x), v_{p,j}(x, y) = \sin(\beta_s x) \sin(\beta_s y) \), are the respective mode shape functions under simply supported boundary conditions. The wave numbers are \( \beta_s = \pi \ell / L_s \) for the string/beam and \( \beta_{s,i,j} = \pi / (L_s + L_p), \beta_{s,j} = j \pi / L_p \) for the plate (the overall wave number denoted as \( \beta_{s,i,j} = \sqrt{\beta_{s,i}^2 + \beta_{s,j}^2} \)). For both the string (\( \kappa = s \)) and the plate (\( \kappa = p \)), the modal displacements \( \bar{u}_{s,i}(t) \) and \( \bar{u}_{p,j}(t) \) are governed by second-order differential equations of the form

\[
m_s \frac{\partial^2 \bar{u}_{s,i}(t)}{\partial t^2} = -k_{s,i} \bar{u}_{s,i}(t) - r_{s,i} \frac{\partial \bar{u}_{s,i}(t)}{\partial t} + \bar{F}_{s,i}(t),
\]

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

\[
m_s = \rho_s A_s L_s, k_{s,i} = \frac{E_s I_s k_s^4}{L_s \gamma_s \beta_s^2}, r_{s,i} = \frac{L_s}{2} \gamma_s \beta_s^2,
\]

\[
m_p = \rho_p h_p L_p L_s, k_{p,l} = \frac{L_s L_p G_p}{4} \beta_s^2, r_{p,l} = \frac{L_s L_p}{4} \gamma_p \beta_s.
\]
The modal frequencies are $\omega_{n,t} = \sqrt{k_n/t_n - \zeta_n, t}$, and the modal attenuation rates are parameterised here in the convenient low-parameter frequency-dependent form

$$\zeta_{n,t} = \frac{r_n}{2m_n} = \sigma_{n,0} + \sigma_{n,1} \alpha + \sigma_{n,2} \beta,$$

(12)

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

$$\vec{F}_n,t(t) = \int_0^{t_n} \psi_n(z) \left[ \psi_n(z) F_t(t) + \psi_n^d(z) F_d(t) \right] dz = g_{n,t} F_1(t) + g_{n,1} F_2(t) + g_{n,1} F_3(t),$$

(13)

$$\vec{F}_p,t(t) = - \int_0^{t_p} \int_0^{t_p} \vec{v}_p(x,y) \vec{\psi}(x,y) F_2(t) dy dx = -g_{p,t} F_2(t),$$

(14)

where (for $v = d, e$)

$$g_{v,t} = v_2(z_v), \quad g_{v,1} = v_3(z_v), \quad g_{v,1} = v_3(x_v, y_v).$$

(15)

For audio output, the plate momenta are modally expanded as

$$\bar{p}_{n,k} = p_{n,k} \sum_{k=0}^{M_p} \frac{a_{n,k}}{\Delta t},$$

(16)

where $a_{n,k} = v_{p,t}(x, y, k) and M_p = M_z M_q$.

### 3. NUMERICAL FORMULATION

#### 3.1. Discretisation in Time

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

$$\frac{\partial \vec{u}_{n,t}}{\partial t} = \frac{\vec{p}_{n,t}}{m_n}, \quad \frac{\partial \vec{p}_{n,t}}{\partial t} = -k_{n,t} \vec{u}_{n,t} - r_{n,t} \vec{u}_{n,t} + \frac{g_{n,t}}{\Delta t} \vec{u}_{n,t} + \frac{g_{n,1}}{\Delta t} \vec{p}_{n,1},$$

(17)

$$\frac{\partial \vec{u}_{0,t}}{\partial t} = \frac{\vec{p}_{0,t}}{m_0}, \quad \frac{\partial \vec{p}_{0,t}}{\partial t} = -r_0 \vec{u}_{0,t} + \vec{F}_1 + \vec{F}_2 + \vec{F}_3,$$

(18)

$$\frac{\partial \vec{u}_{p,t}}{\partial t} = \frac{\vec{p}_{p,t}}{m_p}, \quad \frac{\partial \vec{p}_{p,t}}{\partial t} = -k_{p,t} \vec{u}_{p,t} - r_{p,t} \vec{u}_{p,t} - \vec{g}_{p,t},$$

(19)

where $p_v$ denotes momentum. Gridding time as $u^n = u(n \Delta t)$, the following sum and difference operators

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

(20)

are then employed to discretise these equations at $t = (n + \frac{1}{2}) \Delta t$.

For $F_n^{n+\frac{1}{2}} = \frac{1}{2} \mu F_n^{n+1} + \frac{1}{2} \mu F_n^{n}$, $F_p^{n+\frac{1}{2}} = \frac{1}{2} \mu F_p^{n+1} + \frac{1}{2} \mu F_p^{n}$,

$$\delta \vec{u}_{n,t} = \frac{\vec{p}_{n,t}}{2m_n}, \quad \delta \vec{p}_{n,t} = -k_{n,t} \vec{u}_{n,t} - r_{n,t} \vec{u}_{n,t} + \frac{g_{n,t}}{\Delta t} \vec{u}_{n,t} + \frac{g_{n,1}}{\Delta t} \vec{p}_{n,1},$$

(21)

$$\delta \vec{u}_{0,t} = \frac{\vec{p}_{0,t}}{2m_0}, \quad \delta \vec{p}_{0,t} = -r_0 \vec{u}_{0,t} + \vec{F}_1 + \vec{F}_2 + \vec{F}_3,$$

(22)

$$\delta \vec{u}_{p,t} = \frac{\vec{p}_{p,t}}{2m_p}, \quad \delta \vec{p}_{p,t} = -k_{p,t} \vec{u}_{p,t} - r_{p,t} \vec{u}_{p,t} - \vec{g}_{p,t},$$

(23)

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

$$F_d^{n+\frac{1}{2}} = -r_d \delta u_{d}, \quad F_e^{n+\frac{1}{2}} = k_e \frac{\mu u_t}{2} + \frac{\delta V}{\delta u_t},$$

(24)

the latter utilising the nonlinear spring element potentials

$$V_e(u_t) = \left( \frac{k_e}{\alpha + 1} \right) \lvert u_t \rvert^{\alpha+1} + \left( \frac{k_e}{\alpha + 1} \right) \lvert -u_t \rvert^{\alpha+1}.$$

(25)

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

$$\kappa_{n,t} \rightarrow \kappa_{n,t} = \frac{4m_n \kappa_{n,t}}{\Delta t^2}, \quad r_{n,t} \rightarrow r_{n,t} = \frac{2m_n b_{n,t}}{\Delta t},$$

(26)

where, with $R_{n,t} = \exp(-\kappa_{n,t} \Delta t)$ and $\Omega_{n,t} = \cos(\omega_{n,t} \Delta t)$,

$$b_{n,t} = \frac{1 - 2R_{n,t} \Omega_{n,t} + R_{n,t}^2}{1 + 2R_{n,t} \Omega_{n,t} + R_{n,t}^2},$$

(27)

As explained in previous work [12], this eliminates numerical dispersion and attenuation at the sub-system resonance frequencies. In addition, each modal weight $g_{n,t}$ ($v = n, 1$) has been substituted in (21-23) with $\vec{g}_{n,t} = \vec{W}_{n,t} \left( \frac{1}{\Omega_{n,t}} \right)$, where

$$\vec{W}_n(f) = \left\{ \begin{array}{ll}
\frac{1}{2} & : f < f_n \\
\frac{1}{2} & : f_n \leq f < f_n \\
0 & : f \geq f_n
\end{array} \right.$$
respectively, and with \( b_0 = (r_0 \Delta_0) / (2m_b) \). The other matrices in (31) are

\[
G_c = \begin{bmatrix} g_c & 0_b & 0_p \\ 0_b & 1 & 0_p \\ 0_p & 0 & 0_p \end{bmatrix}, \quad G_e = \begin{bmatrix} g_e & 0_b & 0_p \\ 0_b & 0 & 1 \\ 0_p & 0 & 0_p \end{bmatrix}, \\
G_d = \begin{bmatrix} g_d & 0_b & 0_p \\ 0_b & 0_b & 0 \\ 0_p & 0 & 0_p \end{bmatrix}, \quad \Xi = \begin{bmatrix} \xi_c & 0_b & 0_p \\ 0_b & \xi_c & 0 \\ 0_p & 0 & \xi_p \end{bmatrix},
\]

where matrices \( G_{c,e,d} \) feature modal column vectors \( g_{c,e,d} \) which have \( M \times N \) elements defined by (15). \( \xi_c = \Delta_c / (2m_c) \), and where \( I_{L} \) are \( M_c \times M_c \) identity matrices. By setting \( s = \delta u = \mu q \) from (30), equation (31) can be reworked into

\[
\bar{s} = \bar{e} + H_c \bar{F}^{n+\frac{1}{2}} + H_d \bar{F}^{n+\frac{1}{2}}, \tag{34}
\]

where

\[
e = 2C (q^n - A u^n) + H_c \bar{F}^{n+\frac{1}{2}}, \tag{35}
\]

\[
H_c = \begin{bmatrix} h_c & 0_b & 0_p \\ -\xi_c & \xi_c & 0 \\ 0_b & 0 & 0_p \end{bmatrix}, \quad H_s = \begin{bmatrix} h_s & 0_b & 0_p \\ 0_b & 0 & \xi_p \\ 0_p & \xi_p & 0_p \end{bmatrix}, \tag{36}
\]

\[
H_d = \begin{bmatrix} h_d & 0_b & 0_p \\ 0_b & 0 & 0_p \end{bmatrix}, \quad C = \begin{bmatrix} C_s & 0_b & 0_p \\ 0_b & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0_p \\ 0 & 1 & 0 \end{bmatrix} \tag{37}
\]

Once known, the step vector \( \bar{s} \) is updated to the modal states with

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

after which the output signal vector can be computed as

\[
p^n = W_s \bar{q}^n, \tag{39}
\]

where \( W_s = (2m_p) \Delta_c^{-1} [g_{a,1}, g_{a,2}, \ldots, g_{a,M}]^T \) and where \( g_{a,k} \) is the plate modal column vector for the pickup position (\( k_a, y_a \)). Note that all of the \( M \times M \) matrices \( \{A, B, C, F, G_{c,e,d} \} \) featured above, with \( M = M_e + M_p + 1 \), 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 \( \bar{s} \), we first eliminate the damper force from the system. From (24), one may write

\[
\bar{F}^{n+\frac{1}{2}} = -r_d \Delta_t^{-1} s_d, \tag{40}
\]

where \( s_d = \delta u_d \), which (from multiplying (24) with \( G_d^T \)) also equals

\[
s_d = G_d^T \bar{e} + H_c \bar{F}^{n+\frac{1}{2}} + G_d^T H_d \bar{F}^{n+\frac{1}{2}}. \tag{41}
\]

Eliminating both \( s_d \) and \( \bar{F}^{n+\frac{1}{2}} \) then allows re-writing (24) as

\[
\bar{s} = \bar{w} + \Phi_c \bar{F}^{n+\frac{1}{2}}, \tag{42}
\]

where \( \bar{w} = e - \delta u_0 G_0^T \bar{e} \) and \( \Phi_c = H_c - r_d \delta (G_0^T H_d + \Delta_c) \). Next, the modal coordinate equation (22) is transformed into an equation in the spring connection coordinates by pre-multiplying (22) with \( -G_c \), which gives

\[
\mathbf{M} \bar{F}^{n+\frac{1}{2}} = \mathbf{w} - s, \tag{43}
\]

where \( \mathbf{w} = -G_c^T \bar{w}, s = \delta u (u^e = [u_1^e, u_2^e]^T) \), and

\[
\mathbf{M} = G_c^T \Phi_c = \begin{bmatrix} (\xi_b + \phi_p) & -\xi_b \\ -\xi_b & (\xi_b + \phi_p) \end{bmatrix}, \tag{44}
\]

with \( \phi_p = g_p^T h_c - r_d (g_p^T h_c) \xi_c^p \) and \( \phi_p = g_p^T h_c \). From the second equation in (35), one may write

\[
F^{n+\frac{1}{2}} = \lambda(s) + b_k \left( \frac{1}{2} s + u^n \right), \tag{45}
\]

where \( \lambda(s) = [\lambda_1(s), \lambda_2(s)]^T \) with, for \( \ell = 1, 2 \)

\[
\lambda_\ell(s) = \frac{V(s) + u^{\ell}_n - V(s卐^{\ell}_n)}{s}. \tag{46}
\]

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

\[
\mathbf{M} \lambda(s) + \left( 1 + \frac{1}{2} b_k \mathbf{M} \right) s + \ell = 0, \tag{47}
\]

where \( \ell = b_k \mathbf{M} u^n - w \). These can be solved iteratively using Newton’s method. Once \( s \) is solved, the spring forces are updated from (43), after which the step vector \( \bar{s} \) is calculated with (42). Besides updating the states with (35) and the output with (39), it is also required that the connection displacement vector is updated as \( u^{n+1} = s + u^n \).

### 3.4. Uniqueness and Convergence

Defining \( \lambda'_\ell = \partial \lambda / \partial s \geq 0 \), the Jacobian to be used for solving (47) is

\[
J = \begin{bmatrix} 1 + (\xi_c + \phi_p) \left( \lambda_1' + b_k \right) & -\xi_b \left( \lambda_2' + b_k \right) \\ -\xi_b \lambda_1' + b_k \left( \lambda_2' + b_k \right) & 1 + (\xi_c + \phi_p) \left( \lambda_2' + b_k \right) \end{bmatrix}, \tag{48}
\]

which is clearly positive definite for any possible bridge mass (meaning \( \xi_c \) is finite), hence (47) has a unique root \( s \). \( J \) is also a so-called M-matrix, which is a condition for global convergence [13]. From the fact that \( \lambda_\ell(s_k) \) has a single inflection point (i.e. concave for \( s_k < 0 \) and convex for \( s_k > 0 \)), it then follows that there is guaranteed monotonic convergence from any \( s \) for which each of the elements \( s_k \) has the same sign as the corresponding element of the actual root and satisfies \( |s_k| \geq |s| \) (see [13], 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 I) 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:

$$H^n = \sum_{l=1}^{M_s} \left( \frac{(p_{sl}^n)^2}{2m_s} \right) + \sum_{l=1}^{M_p} \left( \frac{(p_{pl}^n)^2}{2m_p} + \frac{k_{pl}^n(u_{ln}^n)^2}{2} \right) + \frac{1}{2}k_L \left( u_1^2 + u_2^2 \right) + V_1(u_1^n) + V_2(u_2^n).$$

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 = (\mathbf{q}^n)^T \Xi^{-1} \mathbf{q}^n + (\mathbf{u}^n)^T \Xi^{-1} \mathbf{A} \mathbf{u}^n + \frac{1}{2}k_L \left\| \mathbf{u}^n \right\|^2_2 + \| \mathbf{V}(\mathbf{u}^n) \|_2,$n

A numerical power balance that holds under time-invariant parameters is obtained by first pre-multiplying (43) with $\Xi^{-1}$ and, subsequently, the left-hand side with $(\delta \mathbf{u})^T$ and the right-hand side by $(\delta \mathbf{u})^T$, which yields

$$\frac{\Delta H}{\Delta t} = H^{n+1} - H^n = P^{n+\frac{1}{2}} - Q^{n+\frac{1}{2}},$$

where $P^{n+\frac{1}{2}} = \frac{1}{4} \Delta t \cdot \mathbf{G} \mathbf{u} + \mathbf{F} \mathbf{u}^{n+\frac{1}{2}}$ is the input power and $Q^{n+\frac{1}{2}} = \left( \delta \mathbf{u} \right)^T \Xi^{-1} \left( \delta \mathbf{u} \right) + r_n \left( \delta \mathbf{u} \right)^T \delta \mathbf{u} + r_n \left( \delta h_{u1} \right)^2 \geq 0$ (52) is the dissipated power. Hence the power balance, 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 $\Delta t$. 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 EXPANSION

4.1. Control Parameters

Exposing the user to the full set of parameters featuring in (19) 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 $l_s = L_x, l_y = 1$ and fixing the string mass per unity length at $\rho_u A_u = 0.001\text{kg/m}$. The following parameters

<table>
<thead>
<tr>
<th>STRING</th>
<th>Fig 4</th>
<th>Fig 5</th>
<th>Fig 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>fundam. freq. [Hz]</td>
<td>0 &lt; $f_s &lt; \frac{2\pi}{\Delta t}$</td>
<td>100</td>
<td>47.3</td>
</tr>
<tr>
<td>inharmonicity coeff.</td>
<td>$0 \leq B_s$</td>
<td>$10^{-5}$</td>
<td>$10^{-5}$</td>
</tr>
<tr>
<td>damping [s$^{-1}$]</td>
<td>$0 \leq \sigma_s,0$</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>damping [m/s]</td>
<td>$0 \leq \sigma_s,1$</td>
<td>$10^{-3}$</td>
<td>4</td>
</tr>
<tr>
<td>damping [m$^2$/s]</td>
<td>$0 \leq \sigma_s,2$</td>
<td>$10^{-3}$</td>
<td>4</td>
</tr>
<tr>
<td>damper [s$^{-1}$]</td>
<td>$0 \leq \sigma_d$</td>
<td>500</td>
<td>0</td>
</tr>
<tr>
<td>connection position</td>
<td>$0 &lt; \frac{\zeta}{\gamma_{p,k}} &lt; 1$</td>
<td>0.87</td>
<td>0.93</td>
</tr>
<tr>
<td>damper position</td>
<td>$0 &lt; \zeta_{p,k} &lt; 1$</td>
<td>0.99</td>
<td>0.99</td>
</tr>
<tr>
<td>excitation position</td>
<td>$0 &lt; \zeta_{x_{p,k}} &lt; 1$</td>
<td>0.07</td>
<td>0.5</td>
</tr>
</tbody>
</table>

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

$$\bar{f}_s = f_{s,1} = \frac{1}{2} \sqrt{\left( \frac{\rho_s I_s}{\rho_u A_u} \right) \pi^2 + \left( \frac{T_s}{\rho_u A_u} \right)},$$

$$\bar{B}_s = \pi^2 \frac{E_s I_s}{T_s},$$

$$\zeta_{s} = \frac{r_b}{2m_s},$$

$$\zeta_{x} = \frac{r_b}{2m_x},$$

$$\zeta_{y} = \frac{r_b}{2m_y},$$

$$\bar{f}_p = f_{p,1} = \frac{1}{2} \sqrt{\left( \frac{G_{p,2}^2}{\rho_y h_y} \right)} \left( L_x^2 + L_y^2 \right),$$

$$\bar{x}_c = \frac{x_c}{L_x},$$

$$\bar{y}_c = \frac{y_c}{L_y},$$

$$\bar{x}_{x,k} = \frac{x_{x,k}}{L_x},$$

$$\bar{y}_{x,k} = \frac{y_{x,k}}{L_y}.$$

Furthermore, the connection spring stiffness constants in (19) are parameterized as follows:

$$k_{x} = (1 - \eta) k_{b},$$

$$k_{y} = \eta k_{b} G_{p,2} \cdot 10^4(\alpha - 1),$$

where $k_{b} = \frac{1}{2} k_{b} \cdot 10^4(\alpha - 1)$ helps ensuring, in combination with the constraints imposed on the parameters, that the overall stiffness does not exceed
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 $F_b(t) = g_b m_b$, 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 $R_{bs} = 6$), 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 ($R_{bs} = 0.6$) 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 $\eta$ parameter. Fig. 5 shows the spectrograms resulting for $\eta = 0$ (linear springs) and $\eta = 1$ (fully nonlinear springs), with $a = 3$. 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...
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 alongside various further explorative soundings 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 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 up about 5% of the computation when the exit condition chosen as $||s_{i+1} - s_j|| < 10^{-15}$, where $j$ indicates the iteration.

The run-time computations also include re-calculation of system coefficients, such as the non-zero elements of $A$, $C$, $h_s$, and $h_p$, 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 $M_p \approx 4M_s$, 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 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 [13]. 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-
The tunability of the parameters indirectly relies on the modal expansion approach, which - as explained in \[3.1\] - enables eliminating the dispersion and attenuation 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 $G_\kappa$ and $H_\kappa$, and the diagonality of matrices $A$ and $C$, 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 nonsmooth 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 $\delta$. 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 tunability 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