# REAL-TIME IMPLEMENTATION OF THE SHAMISEN USING FINITE DIFFERENCE SCHEMES

**Titas LASICKAS** (tlasic16@student.aau.dk) [1], **Silvin WILLEMSEN** (sil@create.aau.dk) [2], and **Stefania SERAFIN** (sts@create.aau.dk) [2]

[1] *CREATE*, **Aalborg University**, Copenhagen, Denmark
[2] *Multisensory Experience Lab, CREATE*, **Aalborg University**, Copenhagen, Denmark

## ABSTRACT

The shamisen is a Japanese three-stringed lute. It is a chordophone that has the front of the body covered by a tensioned membrane which greatly contributes to the distinct sound of the instrument. Although the shamisen is a traditional Japanese instrument, it is a rare instrument in the rest of the world, making it mostly inaccessible by the majority of artists. To our knowledge, no physically modelled synthesizer of the shamisen is available, forcing producers and musicians to use samples. The objective of this paper is to make the shamisen's distinct sound more accessible to digital music artists. The real-time implementation of the shamisen physical model is presented along with the derivation of solution using the finite-difference time-domain (FDTD) methods. The digital instrument sounds mostly as intended, though lacking the shamisen's distinct buzzing sound requiring further development.

## 1. INTRODUCTION

The shamisen (see Figure 1) is a Japanese three-stringed lute, with origins in China. This instrument is a chordophone that has a membrane covering its soundbox; this stretched membrane contributes to the distinct sound of the instrument. The instrument is played using a bachi, a large plectrum which is held in one hand, while using the fingers of the other hand to pinch the strings against the neck of the instrument, allowing the user to play different pitches. The timbre of the shamisen has a very distinct buzzing which is associated with a low nut which lets a vibrating string come in the contact with the neck [1] and the specific playing technique that increases the percussiveness of the instrument by hitting the body with the bachi during the plucking of the strings [2].

The main goal of this paper is the digitalization and real-time simulation of an instrument that, due to its rarity, is not easily available to the majority of artists. Currently, the only method of obtaining shamisen sounds without having the instrument at hand is by using a sample of the instrument [4] or one of the audio plugins [5–7] which are also based on a sampled shamisen. Usually, a single sample of a note played at a single pitch is pitch-shifted enabling artist to play melodies and chords. Such a method relies on the recordings being done in an anechoic chamber to reduce the effects of the room response. Moreover, due to only usually recorded one note sample, the pitch shifting can introduce artifacts. Even when the sampling is done for all possible pitches on each string, the user is stuck with the way the performer played the instrument during the recording. To rectify these undesirable qualities of the sampling method and to create more scope for skillful articulation when performing, the shamisen can be simulated using a physical model instead.

Figure 1. The shamisen (taken from [3]).

Although several stringed musical instruments have been mostly simulated using digital waveguides [8–10], in this paper we model the shamisen by using Finite Difference Time Domain (FDTD) methods [11]. FDTD methods require developing a full mathematical description of the system. Such a description development uses partial differential equations which are discretized using FDTD methods, yielding finite difference schemes (FDSs). FDTD methods provide better spatial accuracy when the model has frequency-dependent damping and dispersion [12, 13]; in addition, FDTD methods are more flexible as no assumptions are being made about the linearity of the solution [11]. Alternatively, a modal approach – such as in [14, 15] – could be used as it is generally much more efficient. Additionally, modal synthesis for 2D systems appeared in [16, 17], however, to retain generality for control and easier implementation when connecting multiple models, FDTD methods are chosen here . In addition, the nonlinear collision of the bachi with the membrane can be modelled straight-forwardly using FDTD methods. The real-time implementation of similar models, modelled using FDTD approach have been achieved by the authors of [18], where a real-time banjo is recreated using a field programmable gate array and by the authors of [19] where

a real-time implementation of tromba marina is achieved using C++ and the JUCE framework [20]. In this work, we offer a real-time implementation of a physical model of the shamisen based on FDTD methods due to its benefits regarding flexibility and accuracy.

This paper is organized as follows: Section 2 presents the physical models used to model the shamisen and Section 3 shows the discretization of these models. Section 4 shows details of the implementation such as parameter choices, excitation and the output of the system, a high level overview of the calculation order and the graphical user interface and control. Section 5 shows the results regarding output sound and speed of the algorithm and discusses these, while the future work as well as concluding remarks are presented in Section 6.

## 2. MODELING THE SYSTEM

The shamisen can be simplified down to five major parts that make up the whole instrument: the three strings, the bridge and the membrane. The rest of the body is excluded to reduce model complexity which is desirable in order to achieve a model capable of running in real-time. This section will derive partial differential equations (PDEs) in a form

$$\mathcal{L}q = 0, \tag{1}$$

for the mentioned parts of the instrument in isolation. Here, $\mathcal{L}$ is a partial differential operator and $q = q(\boldsymbol{x}, t)$ with time $t$ and spatial coordinate $\boldsymbol{x} \in \mathcal{D}$, where domain $\mathcal{D}$ is one-dimensional for the strings and the bridge, and two-dimensional for the membrane. Moreover, the subscripts 's', 'b' and 'm' indicate that the variables are used for the strings, bridge and the membrane respectively.

### 2.1 Damped Stiff String

Consider a damped stiff string of length $L_\mathrm{s}$ defined over domain $\mathcal{D} = \mathcal{D}_\mathrm{s} = [0, L_\mathrm{s}]$. Recalling Equation (1), the dependent variable $q = u(x, t)$ describes the transverse displacement of the string. Furthermore, the operator $\mathcal{L} = \mathcal{L}_\mathrm{s}$ can be defined as

$$\begin{aligned}\mathcal{L}_\mathrm{s} = \rho_\mathrm{s} A_\mathrm{s} \partial_t^2 - T_\mathrm{s} \partial_x^2 + E_\mathrm{s} I_\mathrm{s} \partial_x^4 \\ + 2\rho_\mathrm{s} A_\mathrm{s} \sigma_{0,\mathrm{s}} \partial_t - 2\rho_\mathrm{s} A_\mathrm{s} \sigma_{1,\mathrm{s}} \partial_t \partial_x^2.\end{aligned} \tag{2}$$

Here, $\partial_t$ and $\partial_x$ indicate a partial differentiation with respect to time and space respectively. Furthermore, various parameters are used to define the string behaviour such as the material density $\rho_\mathrm{s}$, (circular) cross sectional area $A_\mathrm{s} = \pi r^2$, $r$ being the radius of the string, tension of the string $T_\mathrm{s}$, Young's modulus of the string material $E_\mathrm{s}$ and area moment inertia $I_\mathrm{s} = \pi r^4/4$, along with damping coefficients $\sigma_{0,\mathrm{s}}$ and $\sigma_{1,\mathrm{s}}$.

### 2.2 Damped Bridge

The bridge of the shamisen is modeled as a damped linear bar of length $L_\mathrm{b}$ with domain $\mathcal{D}_\mathrm{b} = [0, L_\mathrm{b}]$, and dependent variable $q = v(x, t)$ describing the transverse displacement. The operator $\mathcal{L} = \mathcal{L}_\mathrm{b}$ is similar to $\mathcal{L}_\mathrm{b}$ in Equation (2) without the tension term resulting in the following

$$\begin{aligned}\mathcal{L}_\mathrm{b} = \rho_\mathrm{b} A_\mathrm{b} \partial_t^2 + E_\mathrm{b} I_\mathrm{b} \partial_x^4 \\ + 2\rho_\mathrm{b} A_\mathrm{b} \sigma_{0,\mathrm{b}} \partial_t - 2\rho_\mathrm{b} A_\mathrm{b} \sigma_{1,\mathrm{b}} \partial_t \partial_x^2.\end{aligned} \tag{3}$$

Various parameters are used to define the behaviour of the bridge, such as the material density $\rho_\mathrm{b}$, (rectangular) cross sectional area $A_\mathrm{b} = bH_\mathrm{b}$, $b$ being the width and $H_b$ the thickness of the bridge, the Young's modulus of the bridge material $E_\mathrm{b}$ and the area moment inertia $I_\mathrm{b} = \frac{1}{12} b H_\mathrm{b}^3$, along with damping coefficients $\sigma_{0,\mathrm{b}}$ and $\sigma_{1,\mathrm{b}}$.

### 2.3 Damped stiff membrane

Finally, the membrane covering the instrument's soundbox is modelled as a rectangular damped stiff membrane with side lengths $L_x$ and $L_y$, domain $\mathcal{D} = \mathcal{D}_\mathrm{m} = [0, L_x] \times [0, L_y]$ and dependent variable $q = w(x, y, t)$. The stiffness of the membrane is simulated using a Kirchhoff thin plate stiffness term. Using the 2D Laplacian

$$\Delta \triangleq \partial_x^2 + \partial_y^2, \tag{4}$$

where $\partial_x$ and $\partial_y$ indicate partial differentiation with respect to two spatial dimensions, the operator $\mathcal{L} = \mathcal{L}_\mathrm{m}$ can be defined as:

$$\begin{aligned}\mathcal{L}_\mathrm{m} = \rho_\mathrm{m} H \partial_t^2 - T_\mathrm{m} \Delta + D \Delta\Delta \\ + 2\rho_\mathrm{m} H \sigma_{0,\mathrm{m}} \partial_t - 2\rho_\mathrm{m} H \sigma_{1,\mathrm{m}} \partial_t \Delta.\end{aligned} \tag{5}$$

Again, various parameters are used to define the behaviour of the membrane, such as the material density $\rho_\mathrm{m}$, the membrane thickness $H_\mathrm{m}$, the tension of the membrane $T_\mathrm{m}$, the stiffness parameter $D = E_\mathrm{m} H_\mathrm{m}^3 / 12(1 - \nu^2)$, where $E_\mathrm{m}$ is the Young's modulus of the membrane material and Poisson ratio $\nu$, along with the damping coefficients $\sigma_{0,\mathrm{m}}$ and $\sigma_{1,\mathrm{m}}$.

### 2.4 Boundary Conditions

Something about distributed systems requiring definitions for what happens at the boundaries.

The string is clamped at the boundaries according to

$$u = \partial_x u = 0, \quad \text{where} \quad x = 0, L_\mathrm{s}. \tag{6}$$

Similarly, the membrane is clamped according to

$$w = \mathbf{n} \cdot \nabla w = 0 \tag{7}$$

where $\nabla w$ denotes the gradient of $w$ and $\mathbf{n}$ is a normal to the membrane area at the boundary.

The bridge is free at the boundaries according to

$$\partial_x^2 v = \partial_x^3 v = 0 \quad \text{where} \qquad x = 0, L_\mathrm{b}. \tag{8}$$

### 2.5 The complete system

Until now, only systems in isolation have been considered, i.e., of form (1). To connect the different components, a spatial Dirac delta function $\delta(\boldsymbol{x} - \boldsymbol{x}_\mathrm{c})$ can be used,

which locates the interaction force between two components to the location $\boldsymbol{x}_{\mathrm{c}} \in \mathcal{D}$. The complete system for the shamisen can be written as

$$\begin{cases} \mathcal{L}_{si} u_i &= -\delta(x - x_{si}) F_{\mathrm{t}i}, & \text{(9a)} \\ \mathcal{L}_{\mathrm{b}} v &= \sum_{i=1}^{3} \delta(x - x_{\mathrm{b}i}) F_{\mathrm{t}i} & \text{(9b)} \\ & \quad - \delta(x - x_{\mathrm{bL}}) F_{\mathrm{bL}} - \delta(x - x_{\mathrm{bR}}) F_{\mathrm{bR}}, \\ \mathcal{L}_{\mathrm{m}} w &= \delta(x - x_{\mathrm{mL}}, y - y_{\mathrm{mL}}) F_{\mathrm{bL}} & \text{(9c)} \\ & \quad + \delta(x - x_{\mathrm{mR}}, y - y_{\mathrm{mR}}) F_{\mathrm{bR}}, \end{cases}$$

where subscript $i$ indicates the $i$'th string, $F_{\mathrm{t}i}$ is the force between individual strings and the bridge and $F_{\mathrm{bL}}$ and $F_{\mathrm{bR}}$ are the forces between the bridge and the membrane at the left (L) and right (R) sides of the bridge respectively. The locations $x_{si}$ and $x_{\mathrm{b}i}$ are the locations where the bridge connects to each individual string and vice versa, the 's' subscript denotes that it is a location along a string and the 'b' subscript denotes the location on the bridge. The three strings have one connection each, while the bridge is connected to all three strings. In addition, the locations with subscripts 'bL' and 'mL' correspond to where the membrane connects to the left side of the bridge and vice versa. The same is indicated for the right side using subscripts 'bR' and 'mR'.

## 3. DISCRETIZATION

This section discretizes the full system described in (9) using FDTD methods. These methods subdivide continuous systems (such as described in the previous section) into time samples and grid points in space. The notation used in this section follows [11].

### 3.1 Finite Difference Operators

The first step of implementing FDSs is to define a sampling interval of the continuous system. Time is discretized as $t = nk$, with temporal index $n \in [0, 1, 2...\infty)$ and sampling interval $k = 1/f_{\mathrm{s}}$ where $f_{\mathrm{s}}$ is the sample rate. Space is discretized as $\boldsymbol{x} = \boldsymbol{l}h$, where the dimension of spatial index $\boldsymbol{l}$ is determined by the number of dimensions of the system at hand and $h$ is some spatial sampling interval. Since up-sampling or down-sampling is not needed, $k$ remains the same for all the parts of the model. Once the sampling intervals have been defined, $q(\boldsymbol{x}, t)$ is approximated as a grid function $q_{\boldsymbol{l}}^n$.

Regardless of the dimension of $\boldsymbol{l}$, shift operators can be applied to a grid function. Temporal shift operators are defined as

$$e_{t+} q_{\boldsymbol{l}}^n = q_{\boldsymbol{l}}^{n+1}, \qquad e_{t-} q_{\boldsymbol{l}}^n = q_{\boldsymbol{l}}^{n-1}. \tag{10}$$

Using these shift operators, more complex difference operators can be defined for approximating a first-order time derivative. The forward, backward and centered difference in time operators can be defined as

$$\delta_{t+} := \frac{e_{t+} - 1}{k} \approx \partial_t, \quad \delta_{t-} := \frac{1 - e_{t-}}{k} \approx \partial_t, \quad \delta_{t\cdot} := \frac{e_{t+} - e_{t-}}{2k} \approx \partial_t. \tag{11}$$

where the forward and backward difference are first-order accurate and the centered difference is second-order accurate. Using these first-order difference operators, the second-order difference operator can be defined as a combination of the forward and backward difference operators:

$$\delta_{tt} = \delta_{t+}\delta_{t-} := \frac{e_{t+} - 2 + e_{t-}}{k^2} \approx \partial_t^2. \tag{12}$$

Spatial shift operators in 1D, i.e., $\boldsymbol{l} = l$ can be similarly defined as

$$e_{x+} q_l^n = q_{l+1}^n, \qquad e_{x-} q_l^n = q_{l-1}^n, \tag{13}$$

after which the following first-order difference operators can be defined using the spatial sampling interval $h$

$$\delta_{x+} := \frac{e_{x+} - 1}{h} \approx \partial_x, \quad \delta_{x-} := \frac{1 - e_{x-}}{h} \approx \partial_x, \quad \delta_{x\cdot} := \frac{e_{x+} - e_{x-}}{2h} \approx \partial_x, \tag{14}$$

(following the same orders of accuracy as the first-order time operators) and second-order difference in space operator

$$\delta_{xx} = \delta_{x+}\delta_{x-} := \frac{e_{x+} - 2 + e_{x-}}{h^2} \approx \partial_x^2. \tag{15}$$

The fourth-order spatial derivative operator is obtained by applying operator (15) twice:

$$\delta_{xxxx} = \delta_{xx}\delta_{xx} = \frac{1}{h^4}(e_{t+}^2 - 4e_{t+} + 6 - 4e_{t-} + e_{t-}^2), \tag{16}$$

where a squared shift operator simply means to apply it twice.

The mixed temporal-spatial derivative operator is obtained by applying the backward-time difference operator from (10) to operator (15)

$$\delta_{t-}\delta_{xx} := \frac{e_{x+} - 2 + e_{x-} - e_{t-}(e_{x+} - 2 + e_{x-})}{kh^2} \approx \partial_t \partial_x^2, \tag{17}$$

where a backward difference in time is chosen to keep the system explicit.

Furthermore, in 2D, i.e., $\boldsymbol{l} = l, m$, shift operators are defined as

$$\begin{aligned} e_{x+} q_{l,m}^n &= q_{l+1,m}^n, \quad & e_{x-} q_{l,m}^n &= q_{l-1,m}^n, \\ e_{y+} q_{l,m}^n &= q_{l,m+1}^n, \quad & e_{y-} q_{l,m}^n &= q_{l,m-1}^n, \end{aligned} \tag{18}$$

and finite difference operators can be defined as

$$\begin{aligned} \delta_\Delta = \delta_{xx} + \delta_{yy} \approx \Delta \quad \text{and} \\ \delta_\Delta \delta_\Delta = \delta_{xxxx} + 2\delta_{xx}\delta_{yy} + \delta_{yyyy} \approx \Delta\Delta, \end{aligned} \tag{19}$$

where $\delta_{yy}$ and $\delta_{yyyy}$ are similarly defined to Equations (15) and (16) but using shifting operator $e_{y+}$. Finally, the mixed temporal-spatial operator in 2D is similarly defined as operator (17) using a backward difference in time operator.

It is important to mention that the discrete FDSs derived from the continuous equations such as (9) are an approximation rather than a sampled version of the system.

### 3.2 Discrete Models

In the following, parameters containing a subscript $i$ indicate that these vary between each individual string.

In the case of the strings, we use $x = lh_{\mathrm{s}i}$ to get $u_i(x,t) \approx u_{i,l}^n$, where $l \in [0,\ldots,N_{\mathrm{s}i}]$ and $N_{\mathrm{s}i} = \mathrm{floor}(L_{\mathrm{s}}/h_{\mathrm{s}i})$. All the strings are the same length, $L_{\mathrm{s}}$, but the grid spacing $h_{\mathrm{s}i}$ depends on the individual string parameters. The minimal grid spacing where the solution is stable is calculated using von Neumann stability analysis [11]:

$$h_{\mathrm{s}i} \geq \sqrt{\frac{c_{\mathrm{s}i}^2 k^2 + 4\sigma_{1,\mathrm{s}}k + \sqrt{(c_{\mathrm{s}i}^2 k^2 + 4\sigma_{1,\mathrm{s}}k)^2 + 16\kappa_{\mathrm{s}i}^2 k^2}}{2}}. \tag{20}$$

Here, the wave speed $c_{\mathrm{s}i} = \sqrt{T_{\mathrm{s}i}/\rho_{\mathrm{s}}A_{\mathrm{s}i}}$ and the stiffness $\kappa_{\mathrm{s}i} = \sqrt{E_{\mathrm{s}}I_{\mathrm{s}i}/\rho_{\mathrm{s}}A_{\mathrm{s}i}}$. The closer $h_{\mathrm{s}i}$ is to this condition, the higher the simulation quality will be. The same goes for the conditions given below.

For the bridge we use $x = lh_{\mathrm{b}}$ to get $v(x,t) \approx v_l^n$, where $l \in [0,\ldots,N_{\mathrm{b}}]$ and $N_{\mathrm{b}} = \mathrm{floor}(L_{\mathrm{b}}/h_{\mathrm{b}})$. The grid spacing $h_{\mathrm{b}}$ is defined as

$$h_{\mathrm{b}} \geq \sqrt{2k\left(\sigma_{1,\mathrm{b}} + \sqrt{\sigma_{1,\mathrm{b}}^2 + \kappa_{\mathrm{b}}^2}\right)}, \tag{21}$$

with stiffness parameter $\kappa_{\mathrm{b}} = \sqrt{E_{\mathrm{b}}{H_{\mathrm{b}}}^2/12\rho_{\mathrm{b}}}$.

In the case of the membrane, we use $x = lh_{\mathrm{m}}$ and $y = mh_{\mathrm{m}}$ to get $w(x,y,t) \approx w_{l,m}^n$ where the horizontal index $l \in [0,\ldots,N_x]$ with $N_x = \mathrm{floor}(L_x/h_{\mathrm{m}})$ and vertical index $m \in [0,\ldots,N_y]$ with $N_y = \mathrm{floor}(L_y/h_{\mathrm{m}})$. The membrane is modelled to be square, i.e., $L_x = L_y$, making $N_x = N_y$. The minimal grid spacing $h_{\mathrm{m}}$ is calculated using

$$h_{\mathrm{m}} \geq \sqrt{c_{\mathrm{m}}^2 k^2 + 4\sigma_{1,\mathrm{m}}k + \sqrt{(c_{\mathrm{m}}^2 k^2 + 4\sigma_{1,\mathrm{m}}k)^2 + 16\kappa_{\mathrm{m}}^2 k^2}}, \tag{22}$$

where the membrane wave speed $c_{\mathrm{m}} = \sqrt{T_{\mathrm{m}}/\rho_{\mathrm{m}}H}$ and the stiffness $\kappa_{\mathrm{m}} = \sqrt{D/\rho_{\mathrm{m}}H}$.

In order to discretize the Dirac delta functions found in system (9) we introduce the spreading operator $J(\boldsymbol{x}_{\mathrm{c}})$ that applies the force to the coordinate $\boldsymbol{x}_{\mathrm{c}}$, which is defined as [19]

$$J(\boldsymbol{x}_{\mathrm{c}}) = \begin{cases} \frac{1}{h^d}, & \boldsymbol{l} = \boldsymbol{l}_{\mathrm{c}} = \mathrm{round}(\boldsymbol{x}_{\mathrm{c}}/h), \\ 0, & \text{otherwise.} \end{cases} \tag{23}$$

The rounding function here is used for simplicity, but higher-order spreading functions as found in [11] may be used. In the equation (23) $d$ is a number of the dimensions of domain $\mathcal{D}$ on which $\boldsymbol{x}_{\mathrm{c}}$ is defined. So in the case of the strings and the bridge $d = 1$ and $d = 2$ for the membrane.

### 3.3 Boundary Conditions

Recalling $N_{\mathrm{s}}$, $N_{\mathrm{b}}$, $N_x$ and $N_y$ from Section 3.2, and the finite difference operators from Section 3.1, the strings and the membrane have their boundaries set as clamped which yields boundary conditions for the strings to be

$$u_0^n = \delta_{x+}u_0^n = 0, \quad \text{and} \quad u_{N_{\mathrm{s}}}^n = \delta_{x-}u_{N_{\mathrm{s}}}^n = 0, \tag{24}$$

and the membrane to be

$$\begin{aligned} w_{0,m}^n &= \delta_{x+}w_{0,m}^n = 0, \\ w_{N_x,m}^n &= \delta_{x-}w_{N_x,m}^n = 0, \\ w_{l,0}^n &= \delta_{y+}w_{l,0}^n = 0, \\ w_{l,N_y}^n &= \delta_{y-}w_{l,N_y}^n = 0. \end{aligned} \tag{25}$$

The bridge on the other hand, has a free boundary condition, which is described as

$$\begin{aligned} \delta_{xx}v_0^n = \delta_{xx}\delta_{x\cdot}v_0^n = 0, \quad \text{and} \\ \delta_{xx}v_{N_{\mathrm{b}}}^n = \delta_{xx}\delta_{x\cdot}v_{N_{\mathrm{b}}}^n = 0. \end{aligned} \tag{26}$$

### 3.4 Complete Discrete System

The discretized version of continuous complete system (9) is

$$\begin{cases} \ell_{\mathrm{s}i}u_{i,l}^n &= -J(x_{\mathrm{s}i})F_{\mathrm{t}i}, & \text{(27a)} \\ \ell_{\mathrm{b}}v_l^n &= \sum_{i=1}^{3} J(x_{\mathrm{b}i})F_{\mathrm{t}i} - J(x_{\mathrm{bL}})F_{\mathrm{bL}} & \text{(27b)} \\ &\quad - J(x_{\mathrm{bR}})F_{\mathrm{bR}}, \\ \ell_{\mathrm{m}}w_{l,m}^n &= J(x_{\mathrm{mL}},y_{\mathrm{mL}})F_{\mathrm{bL}} & \text{(27c)} \\ &\quad + J(x_{\mathrm{mR}},y_{\mathrm{mR}})F_{\mathrm{bR}}, \end{cases}$$

where the $\ell$ operators are discretized versions of the partial differential operators $\mathcal{L}$ in system (9) and follow [11] using centered temporal differences for the frequency independent damping terms and backward differences for the frequency dependent damping terms to keep the system explicit (as mentioned in Section (3.1)). Due to the fact that the forces acting on two connected components are equal and opposite due to the rigid connections used, it can be shown that the system is stable as a whole [11].

### 3.5 Coupling

The shamisen model consists of 3 strings, a bridge and a membrane, all coupled together to form a complete instrument (see Figure 2).

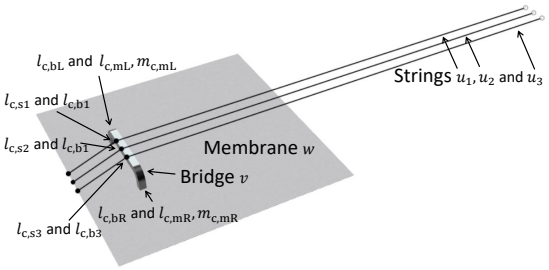


Figure 2. Schematic showing the full coupled system. The different components are highlighted, together with the locations at which they are connected. For the latter, also see Table 1.

The only thing left to do is to calculate the interaction forces between the components. To do this, all schemes in system (27) need to be expanded (written in full) around

every individual connection location which is done by taking an inner product with every individual spreading operator $J$ [11]. Expanding $\ell_{si}$ in Equation (27a) and using subscript $\mathcal{S}_i = i, l_{\mathrm{c},si}$ for compactness, where $i \in [1, 2, 3]$ is the string index and $l_{\mathrm{c},si}$ is the corresponding location on string $i$ where it connects to the bridge, we obtain the following

$$\begin{aligned}\delta_{tt} u^n_{\mathcal{S}_i} = c^2_{si}\delta_{xx} u^n_{\mathcal{S}_i} - \kappa^2_{si}\delta_{xxxx} u^n_{\mathcal{S}_i} - 2\sigma_{0,\mathrm{s}}\delta_{t\cdot} u^n_{\mathcal{S}_i} \\ + 2\sigma_{1,\mathrm{s}}\delta_{t-}\delta_{xx} u^n_{\mathcal{S}_i} - \frac{F_{\mathrm{t}i}}{h_{si}\rho_{\mathrm{s}} A_{si}}\,.\end{aligned} \tag{28}$$

Similarly, for the bridge, we can expand $\ell_{\mathrm{b}}$ from Equation (27b) and take the inner product with the spreading operators belonging to the connections with the strings to obtain

$$\begin{aligned}\delta_{tt} v^n_{l_{ci,\mathrm{b}}} = &-\kappa^2_{\mathrm{b}}\delta_{xxxx} v^n_{l_{ci,\mathrm{b}}} - 2\sigma_{0,\mathrm{b}}\delta_{t\cdot} v^n_{l_{ci,\mathrm{b}}} \\ &+ 2\sigma_{1,\mathrm{b}}\delta_{t-}\delta_{xx} v^n_{l_{ci,\mathrm{b}}} + \frac{F_{\mathrm{t}i}}{h_{\mathrm{b}}\rho_{\mathrm{b}} A_{\mathrm{b}}},\end{aligned} \tag{29}$$

where $l_{ci,\mathrm{b}}$ indicates the location where the bridge connects to string $i$. Taking the inner product with the spreading operators belonging to the connections with the membrane yields

$$\begin{aligned}\delta_{tt} v^n_{l_{\mathrm{c,bL}}} = &-\kappa^2_{\mathrm{b}}\delta_{xxxx} v^n_{l_{\mathrm{c,bL}}} - 2\sigma_{0,\mathrm{b}}\delta_{t\cdot} v^n_{l_{\mathrm{c,bL}}} \\ &+ 2\sigma_{1,\mathrm{b}}\delta_{t-}\delta_{xx} v^n_{l_{\mathrm{c,bL}}} - \frac{F_{\mathrm{bL}}}{h_{\mathrm{b}}\rho_{\mathrm{b}} A_{\mathrm{b}}}, \\ \delta_{tt} v^n_{l_{\mathrm{c,bR}}} = &-\kappa^2_{\mathrm{b}}\delta_{xxxx} v^n_{l_{\mathrm{b,bR}}} - 2\sigma_{0,\mathrm{b}}\delta_{t\cdot} v^n_{l_{\mathrm{c,bR}}} \\ &+ 2\sigma_{1,\mathrm{b}}\delta_{t-}\delta_{xx} v^n_{l_{\mathrm{c,bR}}} - \frac{F_{\mathrm{bR}}}{h_{\mathrm{b}}\rho_{\mathrm{b}} A_{\mathrm{b}}},\end{aligned} \tag{30}$$

where $l_{\mathrm{c,bL}}$ and $l_{\mathrm{c,bR}}$ indicate the locations where the left and right side of bridge connect to the membrane respectively.

Finally, we can follow the same process for the membrane. For brevity, the locations where the bridge and the membrane are connected are defined as $\mathrm{mL} = l_{\mathrm{c,mL}}, m_{\mathrm{c,mL}}$ for the left connection location and $\mathrm{mR} = l_{\mathrm{c,mR}}, m_{\mathrm{c,mR}}$ for the right:

$$\begin{aligned}&\delta_{tt} w^n_{\mathrm{mL}} = c^2_{\mathrm{m}}\delta_{\Delta} w^n_{\mathrm{mL}} - \kappa^2_{\mathrm{m}}\delta_{\Delta}\delta_{\Delta} w^n_{\mathrm{mL}} \\ &- 2\sigma_{0,\mathrm{m}}\delta_{t\cdot} w^n_{\mathrm{mL}} + 2\sigma_{1,\mathrm{m}}\delta_{t-}\delta_{\Delta} w^n_{\mathrm{mL}} + \frac{F_{\mathrm{bL}}}{h^2_{\mathrm{m}}\rho_{\mathrm{m}} H_{\mathrm{m}}}, \\ &\delta_{tt} w^n_{\mathrm{mR}} = c^2_{\mathrm{m}}\delta_{\Delta} w^n_{\mathrm{mR}} - \kappa^2_{\mathrm{m}}\delta_{\Delta}\delta_{\Delta} w^n_{\mathrm{mR}} \\ &- 2\sigma_{0,\mathrm{m}}\delta_{t\cdot} w^n_{\mathrm{mR}} + 2\sigma_{1,\mathrm{m}}\delta_{t-}\delta_{\Delta} w^n_{\mathrm{mR}} + \frac{F_{\mathrm{bR}}}{h^2_{\mathrm{m}}\rho_{\mathrm{m}} H_{\mathrm{m}}}.\end{aligned} \tag{31}$$

All the connection locations where the individual components are coupled together are summarised in Table 1.

There are multiple ways of connecting components together [11]. In this work, rigid connections are assumed, which means that for all $n$

$$\begin{aligned}u^n_{\mathcal{S}_i} = v^n_{l_{ci,\mathrm{b}}}, \quad v^n_{l_{\mathrm{c,bL}}} = w^n_{\mathrm{mL}}, \\ \text{and} \quad v^n_{l_{\mathrm{c,bR}}} = w^n_{\mathrm{mR}}\,.\end{aligned} \tag{32}$$

All the operators in Equations (28), (29) and (30) can be expanded according to the definitions given in Section (3.1) and solved for the states at $n+1$. We can introduce an intermediate state $q^{\mathrm{I}}_l$ which is $q^{n+1}_l$ without the effect of the connection forces. As we know that Equation (32) is true for all $n$, and thus also for $n+1$, we can set the expanded schemes at their connection locations at $n+1$ equal to each other. This yields, for the string-bridge connections

$$u^{\mathrm{I}}_{\mathcal{S}_i} - \frac{F_{\mathrm{t}i}k^2}{h_{si}\rho_{\mathrm{s}} A_{si}(\sigma_{0,\mathrm{s}}k+1)} = v^{\mathrm{I}}_{l_{ci,\mathrm{b}}} + \frac{F_{\mathrm{t}i}k^2}{h_{\mathrm{b}}\rho_{\mathrm{b}} A_{\mathrm{b}}(\sigma_{0,\mathrm{b}}k+1)}, \tag{33}$$

and for the bridge-membrane connections

$$\begin{aligned}v^{\mathrm{I}}_{l_{\mathrm{c,bL}}} - \frac{F_{\mathrm{bL}}k^2}{h_{\mathrm{b}}\rho_{\mathrm{b}} A_{\mathrm{b}}(\sigma_{0,\mathrm{b}}k+1)} = w^{\mathrm{I}}_{\mathrm{mL}} + \frac{F_{\mathrm{bL}}k^2}{h^2_{\mathrm{m}}\rho_{\mathrm{m}} H_{\mathrm{m}}(\sigma_{0,\mathrm{m}}k+1)}, \\ v^{\mathrm{I}}_{l_{\mathrm{c,bR}}} - \frac{F_{\mathrm{bR}}k^2}{h_{\mathrm{b}}\rho_{\mathrm{b}} A_{\mathrm{b}}(\sigma_{0,\mathrm{b}}k+1)} = w^{\mathrm{I}}_{\mathrm{mR}} + \frac{F_{\mathrm{bR}}k^2}{h^2_{\mathrm{m}}\rho_{\mathrm{m}} H_{\mathrm{m}}(\sigma_{0,\mathrm{m}}k+1)}.\end{aligned} \tag{34}$$

These can then be solved for the forces which can finally be substituted back into system (27).

| Strings | $l_{\mathrm{c,s1}}$ | $l_{\mathrm{c,s2}}$ | $l_{\mathrm{c,s3}}$ |
|---|---|---|---|
| Bridge | $l_{\mathrm{c1,b}}$<br>$l_{\mathrm{c,bL}}$ | $l_{\mathrm{c2,b}}$<br>$l_{\mathrm{c,bR}}$ | $l_{\mathrm{c3,b}}$ |
| Membrane | $l_{\mathrm{c,mL}}$, $m_{\mathrm{c,mL}}$ | $l_{\mathrm{c,mR}}$, $m_{\mathrm{c,mR}}$ | |

Table 1. All the connection locations in discrete system (27) (also see Figure 2). Subscripts 'c, s1', 'c, s2' and 'c, s3' indicate the connection points to the bridge on the different strings, 'c1, b', 'c2, b' and 'c3, b' are the corresponding connection points on the bridge for these strings. The subscripts'c, bL' and 'c, bR' indicate connection points on the left the right side of the bridge, in turn, 'c, mL' and 'c, mR' are the same points on the membrane.

## 4. IMPLEMENTATION

The physical model of the shamisen has been implemented in real-time in C++ using the JUCE framework [20]. The code is available at [25]. The control of the digital instrument is limited to the excitation of the separate parts of the instrument including the bridge and the membrane. This section will elaborate on some important considerations regarding the setup of the system, the algorithm and the parameter design. In the end, the graphical user interface (GUI) of the application will be presented.

### 4.1 System setup

The parameters are set at the beginning of the simulation according to Table 2. From this, the spatial sampling intervals $h$ for each individual component are calculated using conditions (20), (21) and (22). After $h_{\mathrm{m}}$ is calculated, we check whether it is smaller than a set minimum value and if it is, use this value instead. Though reducing the quality of the membrane simulation, it increases the computational speed, ultimately allowing the real-time implementation to run smoothly. The value $h_{\mathrm{m,min}} = 0.03$ was heuristically found to be a good trade off between speed and quality.

The spatial sampling intervals are then used to calculate the number of grid points $N_{si}, N_{\mathrm{b}}, N_x$ and $N_y$ which determine the sizes of the state vectors of each component. For every component three vectors (or matrices in the case of the membrane) need to be initialised, saving the states of

| Symbol | Value (Unit) | Parameter |
|---|---|---|
| $b$ | $2.69 \cdot 10^{-3}$ (m)* | Width of the bridge |
| $E_\mathrm{s}$ | $9.9 \cdot 10^{9}$ (Pa)** | String Young's Mod. |
| $E_\mathrm{b}$ | $9.5 \cdot 10^{9}$ (Pa)$^\star$ | Bridge Young's Mod. |
| $E_\mathrm{m}$ | $3 \cdot 10^{9}$ (Pa)$^\star$ | Memb. Young's Mod. |
| $H_\mathrm{b}$ | $7.5 \cdot 10^{-}3$ (m)* | Bridge thickness |
| $H_\mathrm{m}$ | $0.2 \cdot 10^{-}3$ (m)$^{\star\star}$ | Membrane thickness |
| $L_\mathrm{s}$ | 1 (m)$^\dagger$ | String length |
| $L_\mathrm{b}$ | 1 (m)$^\dagger$ | Bridge length |
| $L_x$ | 1 (m)$^\dagger$ | Membrane length |
| $L_y$ | 1 (m)$^\dagger$ | Membrane width |
| $\nu$ | 0.4 | Poisson's ratio |
| $r_{\mathrm{s}1}$ | $4.15 \cdot 10^{-4}$ (m) | Radius string 1 |
| $r_{\mathrm{s}2}$ | $2.83 \cdot 10^{-4}$ (m) | Radius string 2 |
| $r_{\mathrm{s}3}$ | $2.10 \cdot 10^{-4}$ (m) | Radius string 3 |
| $\rho_\mathrm{s}$ | 1156 (kg/m$^3$)** | String density |
| $\rho_\mathrm{b}$ | 500 (kg/m$^3$)$^\star$ | Bridge density |
| $\rho_\mathrm{m}$ | 1150 (kg/m$^3$)$^\star$ | Memb. density |
| | | Frequency independent damping |
| $\sigma_{0,\mathrm{s}}$ | 1.378 (s$^{-1}$) | String damping |
| $\sigma_{0,\mathrm{b}}$ | 1.343 (s$^{-1}$) | Bridge damping |
| $\sigma_{0,\mathrm{m}}$ | 2.756 (s$^{-1}$) | Membrane damping |
| | | Frequency dependent damping |
| $\sigma_{1,\mathrm{s}}$ | $3.57 \cdot 10^{-3}$ (m$^2$/s) | String damping |
| $\sigma_{1,\mathrm{b}}$ | $7.59 \cdot 10^{-2}$ (m$^2$/s) | Bridge damping |
| $\sigma_{1,\mathrm{m}}$ | 0.192 (m$^2$/s) | Membrane damping |
| $T_\mathrm{m}$ | $4 \cdot 10^{3}$ (N/m) | Membrane tension |
| $T_{\mathrm{s}1}$ | 138.67 (N) | String 1 tension |
| $T_{\mathrm{s}2}$ | 145.53 (N) | String 2 tension |
| $T_{\mathrm{s}3}$ | 140.73 (N) | String 3 tension |
| $f_\mathrm{s}$ | $48 \cdot 10^{3}$ (Hz) | Sample rate |

Table 2. List of parameter values used when simulating the shamisen. Parameters were taken from * [21]; ** [22]; $^\star$ [23]; $^{\star\star}$ [24]; all the lengths$^\dagger$ are set to 1 and the other parameters are tuned empirically to produce a desired sound.

$q_l^{n+1}$, $q_l^n$ and $q_l^{n-1}$ respectively. All of these are initialised to 0.

### 4.2 String Tuning

The strings are tuned to 'C4', 'G4' and 'C5' as it is a common tuning for the shamisen. The desired pitch for each string is achieved through empirical testing by mainly changing the tension and adjusting the radius of strings before the start of the simulation. Although the other parameters could be adjusted in order to tune the instrument, changing the density of the string or the Young's modulus would not be possible with a real instrument so it was decided to leave these parameters out when tuning.

### 4.3 Excitation

The system is excited by "plucking" one of the components. Simplified plucking is modeled as a raised cosine added to the current and previous states of the component. 1D components like the string and the bridge use a one dimensional raised cosine, where the membrane uses a 2D version of this as its excitation. A more realistic excitation model is left for future work.

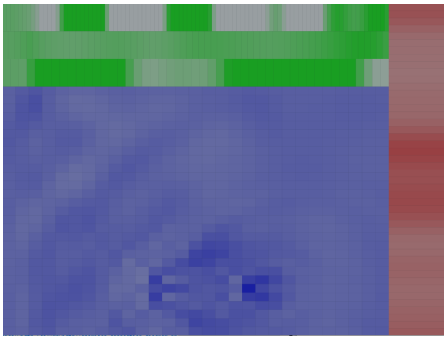

Figure 3. GUI of the digital shamisen. Three green rows at the top are the three strings, red column on the right is the bridge and the blue matrix is the membrane.

### 4.4 Output

The output of the shamisen is a sum of the components' outputs. The strings and the bridge have a single output location which is defined independently for all the components. The membrane output is simplified to be is a sum of the displacements of all the grid points. Depending on the membrane grid size the gain of the output has to be adjusted according to the number of samples in the scheme.

### 4.5 Order of Calculations

The pseudocode shown in Algorithm 1 gives a very high-level overview of the order of the calculations done in the application.

```
Initialise the parameters;
Calculate the number of grid points N;
Initialise the state vectors ;
while The application is running do
    if Mouse click on the component then
        Excite the component;
    end
    Calculate the schemes;
    Calculate the connection forces;
    Add the force to the schemes;
    Retrieve output sound;
    Update the states: q_l^{n-1} = q_l^n;  q_l^n = q_l^{n+1};
end
```

**Algorithm 1:** Pseudocode showing the order of calculations when the program is started and running.

### 4.6 Graphical User Interface

The graphical user interface (GUI) has a simplistic design, where the state variables $q_l^n$ are displayed as shown in Figure 3. Three different colours were chosen, to indicate the three different types of elements. The intensity

of the colour indicates the displacement of $q$ at location $\boldsymbol{l}$. The graphical representation is more suited for showing the normalized vibrations of the different components rather than visualising the instrument itself. Just clicking on the component will excite it and along with the auditory feedback, a graphical representation of what is happening will be shown. The graphics are updated at a rate of 15 Hz.

## 5. RESULTS AND DISCUSSION

Informal listening tests by the authors and some fellow students have confirmed that the shamisen has a specific attack sound that when compared with the recreated shamisen exhibited similar timbral qualities. Naturally, formal listening tests need to be conducted to verify this. The output spectrogram of the three strings excited in succession is visualised in Figure 4.

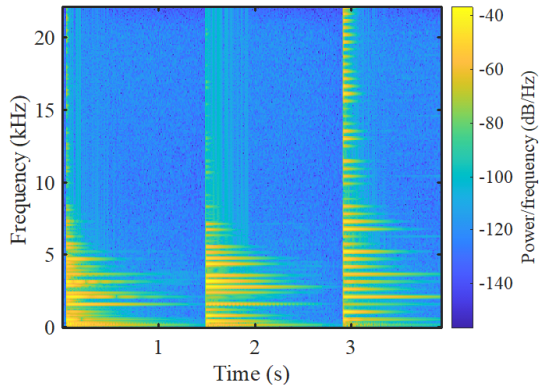


Figure 4. A spectrogram of the three shamisen strings being excited in a succession from the lowest one to the highest. The audio sample used to create this spectrogram is available at [25].

Concerning computational cost, the CPU usage of the real-time application was tested using a MacBook Pro with a 2.2 GHz Intel i7 processor. The strings were continuously excited during the profiling process. The full application uses ca. 51% CPU, whereas the physical model alone without the graphics update uses ca. 39% CPU. This shows that the physical model can easily run in real-time.

As the excitation is adding to the states, sometimes the audio can start clipping, in some cases producing a desirable percussive sound of the membrane and in some cases producing a digital clipping sound, which can affect the timbre of the instrument.

## 6. CONCLUSION AND FUTURE WORK

In this paper, a real-time implementation of a physical model of the shamisen has been presented. The physical model is based on FDTD methods and was implemented in C++ using the JUCE framework. Informal evaluations show that, while lacking the buzzing, the sound has been found natural but not entirely faithful to the real instrument sound.

Future work includes an addition of the buzzing to the instrument timbre which could possibly be achieved by a FDS collision modeling as described in [26] or a modal collisions model by authors of [27]. Furthermore, a comparison between the different sized membranes is needed in order to find the balance of the audio quality and the CPU usage.

Lastly, a physical implementation of the the "fretting" and "plucking" would benefit the model by adding the expressiveness. It would be nice to model "fretting" as done by the authors of [28] and the "plucking" as described in [29–31]. Alternatively a less costly functional transformation method could be applied to model "fretting" [32].

## 7. REFERENCES

[1] Brussels Musical Instruments Museum, "Shamisen," 2021. [Online]. Available: http://www.mim.be/shamisen

[2] B. Ono, "How to play a shamisen/way to play a shamisen is explained here in the picture and moving image." 2009. [Online]. Available: https://www.shamisen.info/ehikikatasan/kotonosoho.html

[3] KogeJoe, "File:heike shamisen.jpg - wikimedia commons," 2011. [Online]. Available: https://commons.wikimedia.org/wiki/File:Heike_shamisen.JPG

[4] J. Champion, "Shamisen instrument samples | free wave samples," 2021. [Online]. Available: https://freewavesamples.com/instrument/shamisen

[5] Syntheway Virtual Musical Instruments, 2021. [Online]. Available: https://syntheway.com/ShamiKoto.htm

[6] Impact Soundworks, "Koto nation by impact soundworks (VST, AU, AAX)," 2021. [Online]. Available: https://impactsoundworks.com/product/koto-nation/

[7] Sonica Instruments, "Sanshin - sonica instruments," 2021. [Online]. Available: http://sonica.jp/instruments/index.php/en/products/sanshin

[8] J. O. Smith, "Physical modeling using digital waveguides," *Computer Music Journal*, 1992.

[9] C. Erkut and V. Välimäki, "Model-based sound synthesis of tanbur, a turkish long-necked lute," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2000, pp. 769–772.

[10] F. Germain and G. Evangelista, "Synthesis of guitar by digital waveguides: Modeling the plectrum in the physical interaction of the player with the instrument," in *2009 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 25–28.

[11] S. Bilbao, *Numerical Sound Synthesis*. John Wiley & Sons, Ltd, 2009.

[12] C. Erkut and M. Karjalainen, "Finite difference method vs. digital waveguide method in string instrument modeling and synthesis," in *Proceedings of the International Symposium on Musical Acoustics (ISMA-02)*, 2002.

[13] ——, "Digital waveguides versus finite difference structures: Equivalence and mixed modeling," *EURASIP Journal on Advances in Signal Processing*, 2004.

[14] M. van Walstijn, J. Bridges, and S. Mehes, "A real-time synthesis oriented tanpura model," in *Proceedings of the 19th International Conference of Digital Audio Effects (DAFx-16)*, 2016.

[15] J. Woodhouse, D. Politzer, and H. Mansour, "Acoustics of the banjo: measurements and sound synthesis," *Acta Acustica*, vol. 5, 2021.

[16] M. Ducceschi and C. Touzé, "Simulations of nonlinear plate dynamics: an accurate and efficient modal algorithm," in *Proceedings of the 18th International Conference on Digital Audio Effects (DAFx-15)*, 2015.

[17] F. Avanzini and R. Marogna, "A modular physically based approach to the sound synthesis of membrane percussion instruments," *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

[18] F. Pfeiffle and R. Bader, "Real-time physical modelling of a complete banjo geometry using FPGA hardware technology," 2009. [Online]. Available: http://systmuwi.de/Pdf/Papers/Bader%20papers/Physical%20Modeling/PhysicalModeling_Banjo/Pfeifle,Bader_04FPGA_Format.pdf

[19] S. Willemsen, S. Serafin, S. Bilbao, and M. Ducceschi, "Real-time implementation of a physical model of the tromba marina," in *Proceedings of the 17th Sound and Music Computing Conference*, 2020.

[20] JUCE, "JUCE," 2020. [Online]. Available: https://juce.com/

[21] "Bachido home," 2021. [Online]. Available: https://bachido.com/

[22] F. Ko, S. Kawabata, M. Inoue, M. Niwa, S. Fossey, and J. Song, "Engineering properties of spider silk fibers," in *MRS Proceedings*, 2001.

[23] "Engineering toolbox," 2001. [Online]. Available: https://www.engineeringtoolbox.com/

[24] F. Azzarto, "What you need to know about...drumheads | modern drummer magazine," 2010. [Online]. Available: https://www.moderndrummer.com/2011/10/what-you-need-to-know-about-drumheads/

[25] T. Lasickas, "Shamisen-juce: Real-time shamisen," 2021. [Online]. Available: https://github.com/titas2001/Shamisen-JUCE

[26] M. Ducceschi and S. Bilbao, "Non-iterative solvers for nonlinear problems: the case of collisions," in *Proceedings of the 22nd International Conference of Digital Audio Effects (DAFx-19)*, 2019.

[27] C. Issanchou, J.-L. Le Carrou, S. Bilbao, C. Touzé, and O. Doaré, "A modal approach to the numerical simulation of a string vibrating against an obstacle: Applications to sound synthesis," in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, 2016.

[28] S. Bilbao and A. Torin, "Numerical modeling and sound synthesis for articulated string/fretboard interactions," in *Journal of the Audio Engineering Society*, vol. 63, 2015.

[29] S. Willemsen, N. Andersson, S. Serafin, and S. Bilbao, "Real-time control of large-scale modular physical models using the sensel morph," in *Proceedings of the 16th Sound and Music Computing Conference*, 2019.

[30] S. Bilbao, M. Ducceschi, and C. Webb, "Large-scale real-time modular physical modeling sound synthesis," in *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, 2019.

[31] P. R. Cook, *Real Sound Synthesis for Interactive Applications*. A K Peters, Ltd, 2002.

[32] L. Trautmann and R. Rabenstein, "Multirate simulations of string vibrations including nonlinear fret-string interactions using the functional transformation method," *EURASIP Journal on Advances in Signal Processing*, 2004.