

## REAL-TIME IMPLEMENTATION OF A PHYSICAL MODEL OF THE TROMBA MARINA

**Silvin Willemsen and Stefania Serafin**  
Multisensory Experience Lab, CREATE  
Aalborg University Copenhagen  
{sil, sts}@create.aau.dk

**Stefan Bilbao and Michele Ducceschi**  
Acoustics and Audio Group  
University of Edinburgh  
{s.bilbao, michele.ducceschi}@ed.ac.uk

### ABSTRACT

The tromba marina is a medieval bowed monochord instrument. The string of the instrument rests on a rattling bridge that, due to the collision with the body, creates a trumpet-like sound. This paper presents a real-time implementation of a physical model of the tromba marina. The goal of the simulation is to make the instrument accessible to a larger audience. The physical model is implemented using finite-difference time-domain (FDTD) methods and non-iterative collision methods. A real-time implementation of the instrument is also presented. The simulation exhibits brass-like qualities and sounds similar to a real tromba marina, but requires further testing to validate the realism.

### 1. INTRODUCTION

The tromba marina (see Figure 1) is a medieval bowed monochord instrument with a long quasi-trapezoidal body and a uniquely fashioned bridge (often called a shoe, because of its shape – see Figure 2). The name of the instrument derives from the fact that *tromba* means *trumpet* in Italian. A peculiarity of the instrument is that a foot of the bridge is free to rattle against the soundboard in sympathy with the vibrating string. This unusual bridge creates a trumpet-like sound. The frequency produced by the instrument is varied by placing the side of the knuckle of the non-dominant hand, lightly, at specific nodal points on the string, in order to select various harmonics of the open string. The dominant hand controls the bow, which is drawn across the string above the non-dominant hand [1].

In this paper, we present a real-time implementation of a physical model of the tromba marina. One of the ultimate goals is the emulation of an instrument that, due to its rarity, is not accessible to a large audience.

Physical modelling for sound synthesis has a long history. Various techniques have been developed to simulate real-world instruments, including mass-spring systems [2], digital waveguides [3] and modal synthesis [4]. Finite-difference time-domain (FDTD) methods were first used for sound synthesis by Hiller and Ruiz in [5–7], later by Chaigne et al. in [8,9] and elaborated upon by Bilbao and colleagues in [10,11]. Compared with other techniques,



Figure 1. The tromba marina from the Danish Music Museum in Copenhagen.



Figure 2. The bridge of the tromba marina from the Danish Music Museum in Copenhagen. The right side is pressed against the body by the string while the left side is free and can rattle against the body.

FDTD methods are more computationally expensive, but easily generalisable and flexible—no assumptions of linearity of travelling wave solutions are employed. Our goal is to implement these techniques in real time and thereby make the simulations playable for the users. For this purpose, we use the expressive Sensel Morph controller [12]. Other work in real-time control of FDTD methods using this controller includes [13].

The emulation of nonlinear collision interactions in musical instruments normally requires the use of iterative solvers (such as the Newton-Raphson algorithm) [14]. For the nonlinear collisions present in the instrument, a method recently proposed in the field of audio by Lopes and Falaize in [15–17] and later by Ducceschi and Bilbao in [18] allows such iterative methods to be sidestepped. It is thus suited to creating a real-time implementation of the tromba marina.

This paper is structured as follows: Section 2 presents the models used and Section 3 shows the discretisation of these. Section 4 provides information about implementation, parameter choices, the graphical user interface and control and mapping. Section 5 shows the results and discusses these. Concluding remarks and future work are presented in Section 6.

## 2. MODELS

The tromba marina can be subdivided into three main components: the string, the bridge and the body. In this section, the partial differential equations (PDEs) of the different components in isolation, under zero-input conditions, will be of the form

$$\mathcal{L}q = 0. \quad (1)$$

Here,  $q = q(x, t)$  represents the state of the component at time  $t$  and spatial coordinate  $x \in \mathcal{D}$ , where the dimensions of domain  $\mathcal{D}$  depend on the component at hand. Furthermore,  $\mathcal{L}$  is a partial differential operator. (Subscripts ‘s’, ‘m’ and ‘p’ used subsequently indicate that (1) applies to the string, bridge (mass) or body (plate), respectively.)

### 2.1 Bowed Stiff String

Consider a damped stiff string of length  $L$  (m), with domain  $\mathcal{D} = \mathcal{D}_s = [0, L]$  and state variable  $q = u(\chi, t)$ . With reference to (1), we define the operator  $\mathcal{L} = \mathcal{L}_s$  as [10]

$$\mathcal{L}_s = \rho_s A \partial_t^2 - T \partial_\chi^2 + E_s I \partial_\chi^4 + 2\rho_s A \sigma_{0,s} \partial_t - 2\rho_s A \sigma_{1,s} \partial_t \partial_\chi^2. \quad (2)$$

Here,  $\partial_t$  and  $\partial_\chi$  indicate partial differentiation with respect to  $t$  and  $\chi$ . The various parameters appear as: material density  $\rho_s$  ( $\text{kg}\cdot\text{m}^{-3}$ ), cross-sectional area  $A = \pi r^2$  ( $\text{m}^2$ ), radius  $r$  (m), tension  $T = (2f_{0,s}L)^2 \rho_s A$  (N),<sup>1</sup> fundamental frequency  $f_{0,s}$  ( $\text{s}^{-1}$ ), Young’s modulus  $E_s$  (Pa), area moment of inertia  $I = \pi r^4/4$  ( $\text{m}^4$ ), and loss coefficients  $\sigma_{0,s}$  ( $\text{s}^{-1}$ ) and  $\sigma_{1,s}$  ( $\text{m}^2/\text{s}$ ). We set the boundary conditions to be simply supported so that

$$u = \partial_\chi u = 0 \quad \text{for} \quad \chi = 0, L. \quad (3)$$

<sup>1</sup> Even though this definition for  $T$  from the fundamental frequency  $f_{0,s}$  is only valid for a simply supported string without stiffness, the effect of the stiffness eventually chosen for  $f_{0,s}$  is negligible.

As the string is excited using a bow, Equation (1) may be augmented as [10]

$$\mathcal{L}_s u = -\delta(\chi - \chi_b) F_b \Phi(v_{\text{rel}}), \quad (4)$$

with externally supplied downward bow force  $F_b = F_b(t)$  (N), spatial Dirac delta function  $\delta(\chi - \chi_b)$  (m) selecting the bow position  $\chi_b = \chi_b(t) \in \mathcal{D}_s$  (m) and dimensionless friction characteristic

$$\Phi(v_{\text{rel}}) = \sqrt{2a} v_{\text{rel}} e^{-av_{\text{rel}}^2 + 1/2}, \quad (5)$$

with free parameter  $a$ . The relative velocity between the string at bow location  $\chi_b$  and the externally supplied bow velocity  $v_b = v_b(t)$  (m/s) is defined as

$$v_{\text{rel}} = \partial_t u(\chi_b, t) - v_b. \quad (6)$$

### 2.2 Bridge

The bridge is modelled as a simple mass-spring-damper system. As this system is point-like, or zero-dimensional, the state variable  $q = w(t)$  and the definition of domain  $\mathcal{D}$  is unnecessary. The operator  $\mathcal{L} = \mathcal{L}_m$  is defined as

$$\mathcal{L}_m = M \frac{d^2}{dt^2} + M \omega_0^2 + M R \frac{d}{dt}, \quad (7)$$

with mass  $M$  (kg), linear angular frequency of oscillation  $\omega_0 = 2\pi f_{0,m}$ , ( $\text{s}^{-1}$ ), fundamental frequency  $f_{0,m}$  ( $\text{s}^{-1}$ ) and damping coefficient  $R$  ( $\text{s}^{-1}$ ).

### 2.3 Body

The body is simplified to a two-dimensional plate with side-lengths  $L_x$  and  $L_y$ , domain  $\mathcal{D} = \mathcal{D}_p = [0, L_x] \times [0, L_y]$  and state variable  $q = z(x, y, t)$ . Using the 2D Laplacian

$$\Delta \triangleq \partial_x^2 + \partial_y^2, \quad (8)$$

the operator  $\mathcal{L} = \mathcal{L}_p$  can be defined as [10]

$$\mathcal{L}_p = \rho_p H \partial_t^2 + D \Delta \Delta + 2\rho_p H \sigma_{0,p} \partial_t - 2\rho_p H \sigma_{1,p} \partial_t \Delta, \quad (9)$$

with material density  $\rho_p$  ( $\text{kg}\cdot\text{m}^{-3}$ ), plate thickness  $H$  (m), stiffness coefficient  $D = E_p H^3/12(1-\nu^2)$ , Young’s modulus  $E_p$  (Pa), dimensionless Poisson’s ratio  $\nu$ , and loss coefficients  $\sigma_{0,p}$  ( $\text{s}^{-1}$ ) and  $\sigma_{1,p}$  ( $\text{m}^2/\text{s}$ ). The boundary conditions of the plate are set to be clamped so that

$$z = \mathbf{n} \cdot \nabla z = 0. \quad (10)$$

where  $\nabla z$  is the gradient of  $z$ , and where  $\mathbf{n}$  indicates a normal to the plate area at the boundary.

### 2.4 Collisions

It can be argued that the greatest contributor to the characteristic sound of the tromba marina is the rattling bridge colliding with the body. A diagram of the bridge with important parts highlighted can be found in Figure 3. A collision can be modelled by including a term to the PDEs mentioned above describing the potential energy of the system

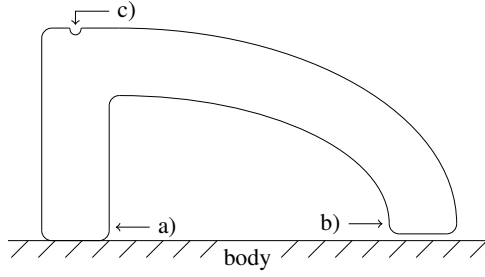


Figure 3. Diagram of the bridge while rattling (view from top of the tromba marina). Indicated are: a) the pivoting point always in contact with the body, b) the rattling point colliding with the body (currently not colliding), and c) the string cavity straight above the middle of the pivoting point.

(further referred to as *the potential*) [18]. For the bridge-body (mass-plate) interaction this potential is defined as follows

$$\phi_{\text{mp}}(\eta_{\text{mp}}) = \frac{K_{\text{mp}}}{\alpha_{\text{mp}} + 1} [\eta_{\text{mp}}]_+^{\alpha_{\text{mp}}+1}, \quad (11)$$

$$K_{\text{mp}} > 0, \quad \alpha_{\text{mp}} \geq 1, \quad \eta_{\text{mp}} \triangleq z(x_{\text{mp}}, y_{\text{mp}}, t) - w(t)$$

where  $K_{\text{mp}}$  is the collision stiffness (N/m if  $\alpha_{\text{mp}} = 1$ ),  $\alpha_{\text{mp}}$  is the dimensionless nonlinear collision coefficient, and  $\eta_{\text{mp}} = \eta_{\text{mp}}(t)$  is the distance between the rattling part of the bridge and the body at the point of collision (m). Furthermore,  $[\eta_{\text{mp}}]_+ = 0.5(\eta_{\text{mp}} + |\eta_{\text{mp}}|)$  is the positive part of  $\eta_{\text{mp}}$ . Note that penalty methods are employed here, where a positive  $\eta_{\text{mp}}$ , i.e., interpenetration of the colliding objects, is intended [19]. The term which can then be included in the PDEs is  $\phi'_{\text{mp}} = d\phi_{\text{mp}}/d\eta_{\text{mp}}$ . As described in [15–18], using this form of the potential requires using iterative methods for solving its discrete counterpart. In [18], the authors propose to rewrite the potential to

$$\psi = \sqrt{2\phi}, \quad (12)$$

and the term included in the PDEs to

$$\phi' = \psi\psi' = \psi \frac{d\psi}{d\eta} \xrightarrow{\text{chain rule}} \psi \frac{\dot{\psi}}{\dot{\eta}}, \quad (13)$$

where the dot above  $\psi$  and  $\phi$  denotes a single time derivative. Equation (13), as can be seen in Section 3, leads to guaranteed stable and explicitly computable simulation algorithms without the need for iterative solvers.

As the string rests on the bridge, the interaction between these components needs to be modelled as well. Even though the bridge-body interaction is perpendicular to the string-bridge interaction, we can model them as being parallel, assuming that a “horizontal” movement of the string causes a “vertical” movement of the rattling part of the bridge. We can use an alternative version of the potential in Equation (11) described in [20] to make the collision two-sided acting as a connection:

$$\phi_{\text{sm}}(\eta_{\text{sm}}) = \frac{K_{\text{sm}}}{\alpha_{\text{sm}} + 1} |\eta_{\text{sm}}|^{\alpha_{\text{sm}}+1}, \quad (14)$$

$$K_{\text{sm}} > 0, \quad \alpha_{\text{sm}} \geq 1, \quad \eta_{\text{sm}} \triangleq w(t) - u(\chi_{\text{sm}}, t)$$

where  $\eta_{\text{sm}} = \eta_{\text{sm}}(t)$  is the distance between the string at the location of the bridge and the bridge itself.

## 2.5 Complete System

A complete system for the tromba marina may be written, in continuous-time as:

$$\begin{cases} \mathcal{L}_s u &= -\delta(\chi - \chi_b) F_b \Phi(v_{\text{rel}}) \\ &+ \delta(\chi - \chi_{\text{sm}}) \psi_{\text{sm}} \psi'_{\text{sm}} \end{cases} \quad (15a)$$

$$\mathcal{L}_m w = -\psi_{\text{sm}} \psi'_{\text{sm}} + \psi_{\text{mp}} \psi'_{\text{mp}}, \quad (15b)$$

$$\mathcal{L}_p z = -\delta(x - x_{\text{mp}}, y - y_{\text{mp}}) \psi_{\text{mp}} \psi'_{\text{mp}}, \quad (15c)$$

$$\eta_{\text{sm}} = w(t) - u(\chi_{\text{sm}}, t), \quad (15d)$$

$$\eta_{\text{mp}} = z(x_{\text{mp}}, y_{\text{mp}}, t) - w(t), \quad (15e)$$

where  $\chi_{\text{sm}} \in \mathcal{D}_s$  is the location of the bridge along the string and  $(x_{\text{mp}}, y_{\text{mp}}) \in \mathcal{D}_p$  is the location on the body with which the bridge collides.

## 3. DISCRETISATION

System (15) is discretised using FDTD methods. These methods subdivide the continuous system in grid points in space and samples in time. Before going into the discretisation of the models, and collision and connection terms in the system described in (15), some finite difference operators are introduced.

### 3.1 Operators

The identity and temporal shift operators are defined as

$$1\eta^n = \eta^n, \quad e_{t+}\eta^n = \eta^{n+1}, \quad e_{t-}\eta^n = \eta^{n-1}. \quad (16)$$

Using these, the operators for the forward, backward and centered time differences can be defined as

$$\delta_{t+} = \frac{e_{t+} - 1}{k}, \quad \delta_{t-} = \frac{1 - e_{t-}}{k}, \quad \delta_t = \frac{e_{t+} - e_{t-}}{2k}, \quad (17)$$

and are all approximations to a first-order time derivative. Furthermore, forwards and backwards averaging operators are defined as

$$\mu_{t+} = \frac{e_{t+} + 1}{2}, \quad \mu_{t-} = \frac{1 + e_{t-}}{2}. \quad (18)$$

and can be used to describe interleaved grid points  $n + 1/2$  and  $n - 1/2$  respectively.

### 3.2 Discrete Models

To approximate the state of a system in isolation we use

$$q(\mathbf{x}, t) \approx q_l^n, \quad (19)$$

where grid function  $q_l^n$  is a discrete approximation to  $q(\mathbf{x}, t)$  at  $t = nk$  with time step  $k$  (s), time index  $n \geq 0$  and grid location  $l$  that depends on domain  $\mathcal{D}$  of the system at hand. In the case of the string, we use  $\chi = lh_s$  with grid spacing  $h_s$  (m),  $l = l \in [0, \dots, N]$  and total number of grid points  $N = L/h_s$  to yield  $u(\chi, t) \approx u_l^n$ .

In the case of the body, we use  $x = lh_p$  and  $y = mh_p$  to get  $z(x, y, t) \approx z_{(l,m)}^n$  where  $\mathbf{l} = (l, m)$  with  $l \in [0, \dots, N_x]$  and  $m \in [0, \dots, N_y]$ . Here,  $N_x = L_x/h_p$  and  $N_y = L_y/h_p$  are the horizontal and vertical number of grid points respectively with grid spacing  $h_p$  (m).

The discretisation of and expansion of operator  $\mathcal{L} \approx \ell$  in the case of stiff strings, mass-spring systems and plates using FDTD methods are well covered in the literature [10] and will not be described in detail in this paper. To obtain the highest accuracy possible while keeping the system explicit (except for the bow), centered differences – which are second-order accurate – have been chosen where possible.

For stability, grid spacings  $h_s$  and  $h_p$  should satisfy the conditions below. In the case of the damped stiff string,

$$h_s \geq \sqrt{\frac{c^2 k^2 + 4\sigma_{1,s} k + \sqrt{(c^2 k^2 + 4\sigma_{1,s} k)^2 + 16\kappa_s^2 k^2}}{2}}, \quad (20)$$

with wave speed  $c = \sqrt{T/\rho_s A}$  and stiffness coefficient  $\kappa_s = \sqrt{E_s I/\rho_s A}$  and in the case of the plate,

$$h_p \geq 2\sqrt{k \left( \sigma_{1,s} + \sqrt{\kappa_p^2 + \sigma_{1,s}^2} \right)}, \quad (21)$$

with stiffness coefficient  $\kappa_p = \sqrt{D/\rho_p H}$ . The closer the grid spacings are to these conditions, the higher the accuracy of the approximation.

In order to discretise the Dirac delta functions found in system (15) we introduce a spreading operator  $J(\mathbf{x}_c)$  that applies a force to coordinate  $\mathbf{x}_c$ , which, in the simplest case, is defined as [10]

$$J(\mathbf{x}_c) = \begin{cases} \frac{1}{h^d}, & \mathbf{l} = \mathbf{l}_c = \text{round}(\mathbf{x}_c/h) \\ 0, & \text{otherwise} \end{cases} \quad (22)$$

Here,  $d$  is the number of dimensions of domain  $\mathcal{D}$  that  $\mathbf{x}$  is defined for, i.e.,  $d = 0$  for the bridge,  $d = 1$  for the string, and  $d = 2$  for the plate. For finer control, a cubic spreading operator  $J_3$  can be introduced [10]. This is used for the bowing term in Equation (4), which is discretised as follows

$$\ell_s u_l^n = -J_3(\chi_b) F_b^n \phi(v_{\text{rel}}^n) \quad (23)$$

where, using the centered difference operator from Equation (17),

$$v_{\text{rel}}^n = \delta_t u_{l_b}^n - v_b^n, \quad (24)$$

with coordinate  $l_b = \chi_b/h_s$ . Equation (24) needs to be calculated using iterative methods.

### 3.3 Collisions using Non-Iterative Methods

For the discrete-time definitions of the potential in (13) we can use

$$\psi \approx \mu_{t+} \psi^{n-1/2} \quad \text{and} \quad \psi' \approx \frac{\delta_{t+} \psi^{n-1/2}}{\delta_t \eta^n}, \quad (25)$$

where  $\psi$  at interleaved grid point  $n - 1/2$  is defined as

$$\psi^{n-1/2} = \mu_{t-} \psi^n. \quad (26)$$

Note that applying a forward or backward difference operator to an interleaved grid – such as  $\delta_{t+} \psi^{n-1/2}$  in Equation (25) – is second-order accurate.

For a system that has a single (upward) collision we get

$$\ell q_l^n = J(\mathbf{x}_c) \left( \mu_{t+} \psi^{n-1/2} \right) \frac{\delta_{t+} \psi^{n-1/2}}{\delta_t \eta^n}. \quad (27)$$

Here, we use the identity

$$\mu_{t+} \psi^{n-1/2} = \frac{k}{2} \delta_{t+} \psi^{n-1/2} - \psi^{n-1/2} \quad (28)$$

and define

$$g^n = \frac{\delta_{t+} \psi^{n-1/2}}{\delta_t \eta^n}, \quad (29)$$

which can be rewritten to

$$\delta_{t+} \psi^{n-1/2} = g^n \delta_t \eta^n. \quad (30)$$

Then, inserting (30) into (28) and this together with (29) into (27) we get

$$\ell q_l^n = J(\mathbf{x}_c) \left( \frac{k}{2} g^n \delta_t \eta^n - \psi^{n-1/2} \right) g^n \quad (31)$$

where  $g^n$  may be explicitly calculated using the analytic expressions for  $\psi$  and  $\phi$  [18]:

$$g^n = \psi' \Big|_{\eta=\eta^n} = \frac{\phi'}{\sqrt{2\phi}} \Big|_{\eta=\eta^n}. \quad (32)$$

Numerical stability of this scheme is shown in [18]. When writing out (32) we can obtain definitions for  $g_{\text{sm}}^n$  using (14)

$$g_{\text{sm}}^n = \text{sgn}(\eta_{\text{sm}}^n) \sqrt{\frac{K_{\text{sm}}(\alpha_{\text{sm}} + 1)}{2}} |\eta_{\text{sm}}^n|^{\frac{\alpha_{\text{sm}} - 1}{2}}, \quad (33)$$

and  $g_{\text{mp}}^n$  using (11)

$$g_{\text{mp}}^n = \sqrt{\frac{K_{\text{mp}}(\alpha_{\text{mp}} + 1)}{2}} [\eta_{\text{mp}}^n]^{\frac{\alpha_{\text{mp}} - 1}{2}}. \quad (34)$$

### 3.4 Complete Discrete System

Introducing for brevity,

$$\xi^n = \frac{k}{2} g^n \delta_t \eta^n - \psi^{n-1/2}, \quad (35)$$

the discrete counterpart of the complete system described in (15) will be

$$\begin{cases} \ell_s u_l^n &= -J_3(\chi_b^n) F_b \Phi(v_{\text{rel}}^n) + J(\chi_{\text{sm}}) \xi_{\text{sm}}^n g_{\text{sm}}^n, & (36a) \\ \ell_m w^n &= -\xi_{\text{sm}}^n g_{\text{sm}}^n + \xi_{\text{mp}}^n g_{\text{mp}}^n, & (36b) \\ \ell_p z_{(l,m)}^n &= -J(x_{\text{mp}}, y_{\text{mp}}) \xi_{\text{mp}}^n g_{\text{mp}}^n, & (36c) \\ \eta_{\text{sm}}^n &= w^n - u_{l_{\text{sm}}}^n, & (36d) \\ \eta_{\text{mp}}^n &= z_{(l_{\text{mp}}, m_{\text{mp}})}^n - w^n, & (36e) \end{cases}$$

where discrete counterparts of connection and collision locations in Equations (36d) and (36e) are described as  $l_{\text{sm}} = \chi_{\text{sm}}/h_s$  and  $(l_{\text{mp}}, m_{\text{mp}}) = (x_{\text{mp}}/h_p, y_{\text{mp}}/h_p)$ . This leaves



us with two different types of update equations, one where  $q_t^{n+1}$  is calculated and one where  $\psi^{n+1/2}$  is calculated.

One might think that due to the centered differences  $\delta_t \eta^n$  still present in Equation (35), our system remains implicit, but as we can insert the definitions for Equations (36d) and (36e) evaluated at the next time index  $n + 1$ , which are already present in  $\ell_s u_l^n$ ,  $\ell_m w^n$  and  $\ell_p z_{(l,m)}^n$ , the Equations in (36) reduce to a system of linear equations that can be solved by a single division.

#### 4. IMPLEMENTATION

The real-time implementation of the system has been done in C++ using the JUCE framework [21] and will be controlled using the Sensel Morph (or simply Sensel) – an expressive touch controller. A demo of the application can be found in [22]. This section will first elaborate some important considerations regarding the setup of the system. Then, the algorithm together with the parameter design will be presented. Finally, the graphical user interface (GUI) will be detailed together with the Sensel and its mapping to the application.

##### 4.1 Introducing an Offset

Firstly, for more realistic and expressive sounds, we model the bridge – and with that, the string – to rest slightly above the body. Expanding  $\ell_m w^n$  in (36b) and including the offset yields

$$\ell_m w^n \Rightarrow M \delta_{tt} w^n + M \omega_0^2 (w^n - w_{\text{off}}) + M R \delta_t w^n \quad (37)$$

where  $w_{\text{off}} \geq 0$  is a predefined offset between the body and the bridge. Furthermore, the second-order time derivative can be defined from the definitions in (17) as

$$\delta_{tt} = \delta_{t+} \delta_{t-} \quad (38)$$

The boundary condition of the string defined in Equation (3) will also change depending on the bridge offset:

$$u = w_{\text{off}} \quad \text{and} \quad \partial_{\chi}^2 u = 0. \quad (39)$$

##### 4.2 Pitch Control

Secondly, as briefly mentioned in Section 1, the way that different pitches are played on the tromba marina, is to slightly rest a knuckle or finger on nodal points along the string to induce harmonics. Thus, a damping finger is implemented. Using the cubic interpolation operator  $I_3$  [10], Equation (36a) can be extended to

$$\ell_s u_l^n = \dots - J_3(\chi_f) I_3(\chi_f) \sigma_f (u_l^n - w_{\text{off}}), \quad (40)$$

where  $0 \leq \sigma_f \leq 1$ ,

which essentially subtracts its own state at location  $\chi_f \in [0, 0.5\chi_{\text{sm}}]$  according to the damping coefficient  $\sigma_f$  (kg·s<sup>-2</sup>) applied. As done in [13], the fractional part used in the spreading operator ( $\alpha_i = \chi_f/h - \text{floor}(\chi_f/h)$ ) is raised to the 7th power as it has been found to scale finger position to pitch more properly in the context of FDTD. As the string is bowed above the damping finger (at the other side

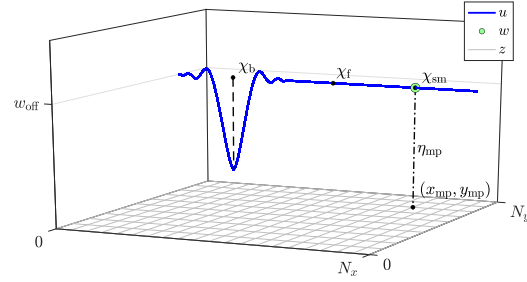


Figure 4. The virtual system in (36) including the offset in Equation (37) and the damping finger in Equation (40), with different important coordinates highlighted. Note that  $\eta_{\text{sm}}$  (Equation (36d)) is not shown as it is close to 0 at all times.

of the rattling bridge) it is essential that the energy from the bow reaches the rattling bridge, which is still the case for lower values of  $\sigma_f$ . A more realistic approach that could be investigated is to model the finger as a mass colliding with the string, rather than imposing the damping directly to the state of the string as presented here.

A schematic plot of the full system, including the offset described in Equations (37) and (39) and the damping finger from Equation (40) can be found in Figure 4.

##### 4.3 Other Considerations

Realistic initialisation of both  $\eta_{\text{sm}}$  and  $\eta_{\text{mp}}$  is essential. In this case (at  $n = 0$ )  $\eta_{\text{sm}}^0 = 0$  and  $\eta_{\text{mp}}^0 \leq 0$  so that no collision is present at initialisation.

After  $h_p$  is calculated in Equation (21), we check whether it is smaller than a set value  $h_{p,\text{min}} = 0.01$ . This reduces the quality of the model, but increases the speed, ultimately allowing for real-time implementation.

##### 4.4 Order of Calculation

The order of calculation is shown in the pseudocode in Algorithm 1. In theory, in order to iteratively calculate the bow force, the collision and connection forces should be included in this. However, as the string is practically never bowed at the bridge position  $\chi_{\text{sm}}$ , these can be calculated independently.

##### 4.5 Parameter Design

The list of parameters used in the implementation can be found in Table 1. As the authors had a real (recreated) tromba marina (presented in [23]) at their disposal, some parameters have been measured in accordance to the real instrument. The others have been tuned by ear by one of the authors.

Regarding the output of the system, through informal testing it was decided to retrieve the output from the state of the plate right at the point of collision  $z_{\text{out}} = (l_{\text{mp}}, m_{\text{mp}})$  combined with the sound of the string at  $u_{\text{out}} = L - \chi_{\text{sm}}$  at a lower volume. It can be argued that the loudest sound comes from the collision between the bridge and the body

```

while application is running do
    1. calculate schemes      ( $\ell q$  in Eqs. (36a-c))
    2. apply bow to string    (Eq. (36a))
    3. apply damping finger   (Eq. (40))
    4. calculate  $g_{sm}^n$  and  $g_{mp}^n$  (Eqs. (33) and (34))
    5. calculate collision and (Eqs. (36a-c))
       connection forces and
       add to schemes
    6. Update states           $q^{n-1} = q^n$ 
                            $q^n = q^{n+1}$ 
                            $\psi^{n-1/2} = \psi^{n+1/2}$ 
end
    
```

**Algorithm 1:** Pseudocode showing the order of calculation after initialisation. Bold symbols denote the collection of states of the entire system ( $q$ ) and potentials ( $\psi$ ).

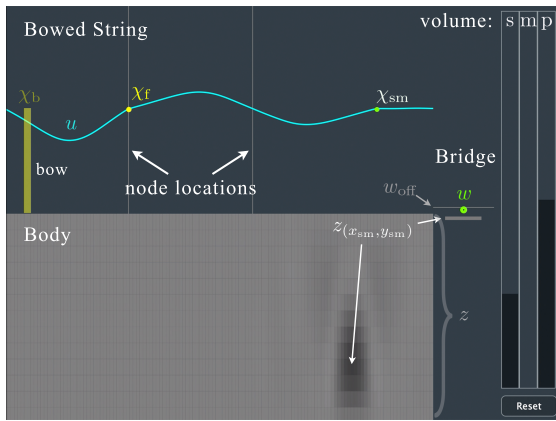


Figure 5. The GUI showing the excited system with components highlighted. A more detailed description can be found in Section 4.6.

making it logical to select this point as the main sound source.

#### 4.6 Graphical User Interface

A screenshot of the GUI is shown in Figure 5. The GUI is divided in four sections, three showing the states of the string, bridge and body respectively and one control section.

Firstly, the string section shows the state of the string  $u$  as a cyan-coloured path and the bow as a yellow rectangle with bow position  $\chi_b$  and its opacity depending on the bow force  $F_b$ . Furthermore, the bridge state  $w$  is shown as a green circle at location (of the bridge along the string)  $\chi_{sm}$ . Finally, the position of the damping finger  $\chi_f$  is displayed as a yellow circle, the size of which depends on damping coefficient  $\sigma_f$ . The position of the finger triggers lines showing the locations of the closest nodes along the

Name	Symbol (unit)	Value
<b>String</b>		
Length	$L$ (m)	1.90*
Material density	$\rho_s$ (kg·m <sup>-3</sup> )	7850
Radius	$r$ (m)	0.0005
Fundamental freq.	$f_0$ (s <sup>-1</sup> )	32*
Young's modulus	$E_s$ (Pa)	$2 \cdot 10^{11}$
Freq. indep. loss	$\sigma_{0,s}$ (s <sup>-1</sup> )	0.1
Freq. dep. loss	$\sigma_{1,s}$ (m <sup>2</sup> /s)	0.05
<b>Bow</b>		
Bow force	$F_b$ (N)	$0 \leq F_b \leq 0.1$
Bow velocity	$v_b$ (m/s)	$-0.5 \leq v_b \leq 0.5$
Free parameter	$a$ (-)	100
<b>Bridge</b>		
Mass	$M$ (kg)	0.001
Fundamental freq.	$f_{0,m}$ (s <sup>-1</sup> )	500
Damping	$R$ (s <sup>-1</sup> )	0.05
<b>Body</b>		
Length	$L_x$ (m)	1.35*
Width	$L_y$ (m)	0.18*
Material density	$\rho_p$ (kg·m <sup>-3</sup> )	50
Thickness	$H$ (m)	0.01
Young's modulus	$E_p$ (Pa)	$2 \cdot 10^5$
Poisson's ratio	$\nu$ (-)	0.3
Freq. indep. loss	$\sigma_{0,p}$ (s <sup>-1</sup> )	2
Freq. dep. loss	$\sigma_{1,p}$ (m <sup>2</sup> /s)	0.05
Min. grid spacing	$h_{p,min}$ (m)	0.01
<b>String-bridge connection</b>		
Stiffness coefficient	$K_{sm}$ (N/m)	$5 \cdot 10^6$
Nonlin. col. coeff.	$\alpha_{sm}$ (-)	1
Bridge location	$\chi_{sm}$ (m)	1.65*
<b>Bridge-body collision</b>		
Stiffness coefficient	$K_{mp}$ (N/m)	$5 \cdot 10^8$
Nonlin. col. coeff.	$\alpha_{mp}$ (-)	1
Bridge location	$(x_{mp}, y_{mp})$ (m,m)	(1.08, 0.135)*
<b>Other</b>		
Offset	$w_{off}$ (m)	$5 \cdot 10^{-6}$
Damp. finger coeff.	$\sigma_f$ (kg·s <sup>-2</sup> )	$0 \leq \sigma_f \leq 1$
Output loc. string	$u_{out}$ (m)	$L - \chi_{sm}$
Output loc. body	$z_{out}$ (m, m)	$(x_{mp}, y_{mp})$

Table 1. List of parameter values used for the simulation. \*These values have been taken from a real (recreated) tromba marina [23].

string according to the following equation

$$\chi_{node}^i = \frac{i \cdot \chi_{sm}}{n} \quad \text{for } i = [1, \dots, n-1], \quad (41)$$

where  $n = \text{round}(\chi_{sm}/\chi_f)$  is an integer closest to the ratio between the string length until the bridge location and the damping finger position. These lines are drawn to help the user place the damping finger at nodes along the string.

Secondly, the bridge section shows the displacement of the bridge  $w$  as a green circle, the state of the body at the collision location  $z(\chi_{sm}, m_{sm})$ , both moving vertically according to their respective displacements and finally, a static grey horizontal line denoting the offset  $w_{off}$ , i.e., the resting position of the bridge.

Thirdly, the body section shows the state of the body  $z$  as a grid of rectangles changing (grey-scale) colour according to their displacement.

Finally, the control section contains three sliders that control the volume-levels of the string (s), bridge (m) and body

(p) respectively (for experimentation of volume ratios between the components) and a reset button to re-initialise the system.

#### 4.7 Sensel Morph and Mapping

The Sensel is an expressive touch controller using ~20,000 pressure-sensitive sensors laid out in an hexagonal grid [12]. It retrieves x and y-positions and pressure at a rate of 150 Hz from which velocities and accelerations can be obtained.

The first finger registered by the Sensel is mapped to the bow: x-position is mapped to bow position  $x_b$ , y-velocity to bow velocity  $v_b$  (y-position is shown in the GUI but does not influence the model directly) and pressure to bow force  $F_b$ . The second finger is mapped to the damping finger: x-position is mapped to finger location  $x_f$  and pressure to damping coefficient  $\sigma_f$ .

### 5. RESULTS AND DISCUSSION

Informal listening by the authors has confirmed that the sound has brass-like qualities and comparison with the recreated tromba marina showed that the sound exhibited similar qualities. Naturally, formal listening tests need to be conducted to verify this.

Disabling the graphics of the application, its CPU usage is 68.9% on a MacBook Pro with a 2.2 GHz Intel i7 processor, easily allowing it to work in real-time. As the heaviest part of the algorithm is the calculation of the body, the minimum grid spacing  $h_{p,min}$  could be set to a higher value to decrease the CPU usage. However, as mentioned, this will decrease the quality of the output sound.

Through using the application, the authors found some odd behaviour, where the bridge ‘gets stuck’ behind the plate, i.e., values for  $\psi_{mp}$  would be negative for a short period of time (one to several samples). The explicit technique used in this work allows for this to happen (and can be proven to still be stable in this case [18]), but it is ‘unphysical’ to have a negative potential as this implies a ‘pulling’ collision. As can be seen from Table 1, the non-linear collision coefficients  $\alpha_{sm}$  and  $\alpha_{mp}$  are set to 1. When increasing these values, this behaviour would arise much more often, and even occur for a prolonged period of time (several seconds to indefinitely). This is also the reason why the reset button presented in Section 4.6 has been implemented. As mentioned in [18], oversampling increases the accuracy of the explicit collision method, and could be a solution to this issue. However, in order for the application to run in real time, this solution can not be afforded without decreasing the quality of the implementation, e.g. increasing  $h_{p,min}$ . Further investigation will be necessary to solve this issue without oversampling.

Lastly, it has been found that when  $|z_{(l,m)}^n| \lesssim 10^{-306}$  (but non-zero) for any coordinate  $(l, m)$  (which happens when the body has not been collided with for a prolonged period of time), the CPU usage increases considerably. This could be explained by the fact that calculations with extremely small values are handled differently by the application. This is solved by implementing a limit to how small

a value for  $z_{(l,m)}^n$  can be. If the value of  $z_{(l_{sm}, m_{sm})}^n$  is lower than this limit, the total plate state is set to 0.

### 6. CONCLUSION AND FUTURE WORK

In this paper, a real-time implementation of a simulation of the tromba marina has been presented. The output sound has been found natural and brass-like by the authors and exhibited similar qualities when compared to a real (recreated) tromba marina.

Future work includes a comparison between the non-iterative methods used in this paper and iterative methods (such as and Newton-Raphson) both regarding algorithm speed and sound quality.

Lastly, for a more physical implementation of the damping finger, it would be good to model it as another mass colliding with the string rather than directly imposing damping onto the string state.

#### Acknowledgments

The authors would like to thank Peter Williams for his valuable feedback on our application.

This work is supported by NordForsk’s Nordic University Hub Nordic Sound and Music Computing Network NordicSMC, project number 86892. Ducceschi’s work was supported by an Early Career Fellowship from the Leverhulme Trust.

### 7. REFERENCES

- [1] D. Munrow, *Instruments of the Middle Ages and Renaissance*. Oxford University Press, USA, 1976.
- [2] C. Cadoz, “Synthèse sonore par simulation de mécanismes vibratoires,” Ph.D. dissertation, Grenoble INP, 1979.
- [3] J. O. Smith, “Physical modeling using digital waveguides,” *Computer music journal*, vol. 16, no. 4, pp. 74–91, 1992.
- [4] J. D. Morrison and J.-M. Adrien, “Mosaic: A framework for modal synthesis,” *Computer Music Journal*, vol. 17, no. 1, pp. 45–56, 1993.
- [5] P. Ruiz, “A technique for simulating the vibrations of strings with a digital computer,” Master’s thesis, University of Illinois, 1969.
- [6] L. Hiller and P. Ruiz, “Synthesizing musical sounds by solving the wave equation for vibrating objects: Part I,” *Journal of the Audio Engineering Society*, vol. 19, no. 6, pp. 462–470, 1971.
- [7] —, “Synthesizing musical sounds by solving the wave equation for vibrating objects: Part II,” *Journal of the Audio Engineering Society*, vol. 19, no. 7, pp. 542–550, 1971.
- [8] A. Chaigne, “On the use of finite differences for musical synthesis. Application to plucked stringed instruments,” *Journal d’Acoustique*, vol. 5, no. 2, pp. 181–211, 1992.

- [9] A. Chaigne and A. Askenfelt, “Numerical simulations of struck strings. I. A physical model for a struck string using finite difference methods,” *Journal of Acoustical Society of America*, vol. 95, no. 2, pp. 1112–1118, 1994.
- [10] S. Bilbao, *Numerical sound synthesis: finite difference schemes and simulation in musical acoustics*. John Wiley & Sons, 2009.
- [11] S. Bilbao, B. Hamilton, R. Harrison, and A. Torin, “Finite-difference schemes in musical acoustics: A tutorial,” *Springer handbook of systematic musicology*, 2018.
- [12] Sensel Inc. (2020) Sensel morph. [Online]. Available: <https://sensel.com/>
- [13] S. Willemsen, N. Andersson, S. Serafin, and S. Bilbao, “Real-time control of large-scale modular physical models using the sensel morph,” *Proc. of the 16th Sound and Music Computing (SMC) Conference*, pp. 275–280, 2019.
- [14] S. Bilbao, A. Torin, and V. Chatziioannou, “Numerical modeling of collisions in musical instruments,” *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2015.
- [15] N. Lopes, T. Hélie, and A. Falaize, “Explicit second-order accurate method for the passive guaranteed simulation of port-hamiltonian systems,” *Proc. 5th IFAC*, 2015.
- [16] A. Falaize and T. Hélie, “Passive guaranteed simulation of analog audio circuits: A port-hamiltonian approach,” *Applied Sciences*, vol. 6, pp. 273–273, 2016.
- [17] A. Falaize, “Modélisation, simulation, génération de code et correction de systèmes multi-physiques audios: Approche par réseau de composants et formulation hamiltonienne à ports,” Ph.D. dissertation, Université Pierre et Marie Curie, Paris, 2016.
- [18] M. Ducceschi and S. Bilbao, “Non-iterative solvers for nonlinear problems: The case of collisions,” *Proc. of the 22th Int. Conf. on Digital Audio Effects (DAFx-19)*, 2019.
- [19] S. Bilbao, A. Torin, and V. Chatziioannou, “Numerical modeling of collisions in musical instruments,” *Acta Acustica united with Acustica*, vol. 101, no. 1, pp. 155–173, 2014.
- [20] S. Bilbao and M. Ducceschi, “Large-scale real-time modular physical modeling sound synthesis,” *Proc. of the 22th Int. Conf. on Digital Audio Effects (DAFx-19)*, 2019.
- [21] JUCE ROLI. (2020) JUCE. [Online]. Available: <https://juce.com/>
- [22] S. Willemsen. (2020) Virtual tromba marina - sensel morph. [Online]. Available: <https://www.youtube.com/watch?v=x72Xh-nUoVc>
- [23] A. Baldwin, T. Hammer, E. Peciulis, P. Williams, D. Overholt, and S. Serafin, “Tromba moderna: A digitally augmented medieval instrument,” *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, vol. 16, pp. 14–19, 2016.