Connected Physical Models controlled with the Sensel Morph

Silvin Willemsen, Nikolaj Andersson and Stefania Serafin

Multisensory Experience Lab, CREATE, Aalborg University Copenhagen
{sil, nsa, sts}@smcnetwork.org

ABSTRACT

Lorum Ipsum

1. INTRODUCTION

The behaviour of musical instruments can be well defined by partial differential equations (PDEs).

Finite-difference schemes (FDSs)

The physical models (PMs) used as a case study in this project are the stiff string and the plate.

This paper is structured as follows: Section 2 will present the PMs used in our implementation, Section 3 will show the how to implement the PMs, Section 2.3

2. MODELS

In this section, the partial differential equations for the damped stiff string and the plate will be presented.

2.1 Stiff string

The state $u = u(x,t)$ describes the transverse displacement of the string. The subscript for u denotes a single derivative with respect to time t or space x respectively. The partial differential equation for the damped stiff string is defined as [1]

$$u_{tt} = \gamma^2 u_{xx} - \kappa^2 u_{xxxx} - 2\sigma_0 u_t + 2\sigma_1 u_{txx}, \quad (1)$$

where γ is wave-speed [m/s], κ is stiffness [?] and $\sigma_0 \geq 0$ and $\sigma_1 \geq 0$ are frequency-dependent and frequency-independent damping respectively.

We can add an excitation term to extend Equation (1) to a bowed string [1]

$$u_{tt} = ... - \delta(x - x_\text{B})F_\text{B}\phi(v_\text{rel}) \quad \text{where} \quad (2)$$

$$v_\text{rel} = u_t(x_\text{B}) - v_\text{B}, \quad (3)$$

where F_B is the bowing force [N], v_B is the bowing velocity [m/s], v_rel is the relative velocity, defined as the difference between the velocity of the string at bowing point x_B and the bowing velocity v_B [m/s] and ϕ is a friction characteristic, which has been chosen to be [1]

$$\phi(v_\text{rel}) = \sqrt{2a}v_\text{rel}\exp\left(-av_\text{rel}^2 + 1/2\right). \quad (4)$$

2.2 Plate

In the case of a plate, the state $u = u(x,y,t)$ is now defined over two spatial dimensions. The PDE for a damped plate is [1]

$$u_{tt} = -\kappa^2\Delta\Delta u - 2\sigma_0 u_t + 2\sigma_1 \Delta u_t, \quad (5)$$

where Δ represents the 2D Laplacian (also see Equation (15)). Just like in the case of the string, an extra term can be added as an input:

$$u_{tt} = ... + F_\text{e}E_\text{e}, \quad (6)$$

where $F_\text{e} = f_\text{e}/\rho AL$ [m/s^2], with bowing force f_e [N], density ρ [kg/m^3], cross-sectional area A [m^2] and string length L [m], and E_e are an excitation function and the excitation area respectively.

2.3 Connections

Adding connections between different PMs, further referred to as elements, adds another term to Equation (2) or (6)

$$u_{tt} = ... + F_\alpha E_\alpha, \quad (7)$$

$$u_{tt} = ... + F_\beta E_\beta, \quad (8)$$

where F_α and F_β are the forces of the connection at connection areas E_α and E_β respectively. If the a connection area consists of only one point, E reduces to $\delta(x - x_c)$ where x_c is the point of connection. We use the implementation as presented in [2] where the connection between two elements is a non-linear spring. The forces it imposes on the elements it connects - denoted by α and β - are defined as

$$F_\alpha = -\omega_0^2\eta - \omega_1^4\eta^3 - 2\sigma_\times\eta_t, \quad (9)$$

$$F_\beta = -M_{\alpha/\beta}F_\alpha, \quad (10)$$

where ω_0 and ω_1 are the linear and non-linear spring coefficients respectively, $\sigma_\times$ is the damping factor, $M_{\alpha/\beta}$ is the mass ratio between the two elements and η is the relative displacement between the connected elements at the point of connection. The subscript t again denotes a time derivative.

3. FINITE-DIFFERENCE SCHEMES

To be able to digitally implement the continuous equations mentioned in the previous section, they need to be approximated. The models can be discretised at times $t = nk$, where $n \in \mathbb{W}$ and $k = 1/f_s$ is the time-step with sample-rate f_s and locations $x = lh$, where $l \in [0, N]$ with N being the total number of points and h is the grid-spacing of the model which is calculated differently for each model (see sub-sections below). The discretised variable u_l^n is $u(x,t)$ at the nth time step and the lth point on the string. In the case of a plate, the second spatial dimension is discretised using $x = lh$ where $l \in [0, N_x]$ with N_x being the total horizontal number of points and $y = mh$ where $m \in [0, N_y]$ with N_y being the total vertical number of points. Approximations for the derivatives in the equations found in Section 2 are described in the following way:

When approximating the PDEs shown in Section 2, we use operators

$$\delta_{xx}u_l^n = \frac{1}{h^2}\big(u_{l+1}^n - 2u_l^n + u_{l-1}^n\big), \tag{11}$$

$$\delta_{t-}u_l^n = \frac{1}{k}\big(u_l^n - u_l^{n-1}\big), \tag{12}$$

$$\delta_{t\cdot}u_l^n = \frac{1}{2k}\big(u_l^{n+1} - u_l^{n-1}\big), \tag{13}$$

$$\delta_{tt}u_l^n = \frac{1}{k^2}\big(u_l^{n+1} - 2u_l^n + u_l^{n-1}\big), \tag{14}$$

$$\delta_{\Delta}u_{l,m}^n = \frac{1}{h^2}(u_{l,m+1}^n + u_{l,m-1}^n + u_{l+1,m}^n + \tag{15}$$

$$u_{l-1,m}^n - 4u_{l,m}^n), \tag{16}$$

3.1 Stiff String

Equation (2) can be approximated using

$$\begin{aligned}\delta_{tt}u_l^n =& \gamma^2\delta_{xx}u_l^n - \kappa^2\delta_{xx}\delta_{xx}u_l^n - 2\sigma_0\delta_{t\cdot}u_l^n \\ &+ 2\sigma_1\delta_{t-}\delta_{xx}u_l^n - \delta_{x_\mathrm{B}}F_\mathrm{B}\phi(v_\mathrm{rel}),\end{aligned} \tag{17}$$

where $\delta_{x_\mathrm{B}} = \delta(x - x_\mathrm{B})$ is the spatial Dirac delta function (1 at x_B, else 0). For our implementation, clamped boundary conditions were used, defined as:

$$u = u_x = 0 \quad \text{where} \quad l = \{0, N\}. \tag{18}$$

For stability reasons, the grid-spacing needs to abide the following condition

$$h \geq \sqrt{\frac{\gamma^2k^2 + 4\sigma_1 k + \sqrt{(\gamma^2k^2 + 4\sigma_1 k)^2 + 16\kappa^2k^2}}{2}}. \tag{19}$$

The closer h is to this limit, the higher the quality of the implementation. The number of points N can then be calculated using

$$N = h^{-1}. \tag{20}$$

3.2 Plate

Equation (6) can be approximated using

$$\begin{aligned}\delta_{tt}u_{l,m}^n =& -\kappa^2\delta_\Delta\delta_\Delta u_{l,m}^n - 2\sigma_0\delta_{t\cdot}u_{l,m}^n \\ &- 2\sigma_1\delta_t\delta_\Delta u_{l,m}^n + F_\mathrm{e}E_\mathrm{e}\end{aligned} \tag{21}$$

In the case of the plate, we set the number of horizontal and vertical points and calculate grid spacing h from that using

$$h = \frac{\sqrt{N_x/N_y}}{N_x}. \tag{22}$$

3.3 Connections

$$F_\alpha = -\omega_0^2\mu_{t\cdot}\eta - \omega_1^4\eta^2\mu_{t\cdot}\eta - 2\sigma_\times\delta_{t\cdot}\eta, \tag{23}$$

$$F_\beta = -M_{\alpha/\beta}F_\alpha, \tag{24}$$

The relative displacement η between α and β can be calculated as

$$\eta^n = h_\alpha u_{\alpha,x_\alpha}^n - h_\beta u_{\beta,x_\beta}^n, \tag{25}$$

which, in other words, is the difference between the state of element α at connection point x_α and the state of element β at connection point x_β scaled by their respective grid-spacings h_α and h_β.

4. PITCH

We implemented a damping point in the model that acts as a finger on the (virtual) neck of the instrument controlling pitch. After u^{n+1} is calculated, the following operation is performed at the point of damping:

$$u_{x_\mathrm{f}}^{n+1} = u_{x_\mathrm{f}}^{n+1}\sigma_\mathrm{f}, \tag{26}$$

where $\sigma_\mathrm{f} \in [0, 1]$ is the damping coefficient of the finger.

5. IMPLEMENTATION

For prototyping `Matlab` was used. We then used C++ along with the JUCE framework for implementing the objects and connections in real-time.

5.1 String

In order to implement the FDSs presented in Section 3 they need to be solved for u^{n+1}. In the case of the string, we obtain

$$\begin{aligned}(1 + \sigma_0 k)u_l^{n+1} =& 2u_l^n - (1 - \sigma_0 k - 2\phi)u_l^{n-1} \\ &+ (\lambda^2 + \phi)(u_{l+1}^n - 2u_l^n + u_{l-1}^n) \\ &- \mu^2(u_{l+2}^n - 4u_{l+1}^n + 6u_l^n - 4u_{l-1}^n + u_{l-2}^n) \\ &+ \phi(u_{l+1}^{n-1} + u_{l-1}^{n-1}) \\ &- k^2\delta_{l_\mathrm{e}}F_\mathrm{e},\end{aligned} \tag{27}$$

where

$$\lambda = \frac{\gamma k}{h}, \quad \mu = \frac{\kappa k}{h^2}, \quad \phi = \frac{2\sigma_1 k}{h^2} \quad \text{and} \quad \delta_{l_\mathrm{e}} = \delta(x - x_{l_\mathrm{e}}).$$

The wave-speed of the string is proportional to the fundamental frequency of the stiff string according to

$$\gamma = 2f_0, \tag{28}$$

and the stiffness can be calculated using

$$\kappa = \frac{\sqrt{B}\gamma}{\pi}, \tag{29}$$

where $B = 0.001$ is the inharmonicity coefficient [m^{-2}]. The output is retrieved at $l = \text{floor}(0.75N)$ for all strings.

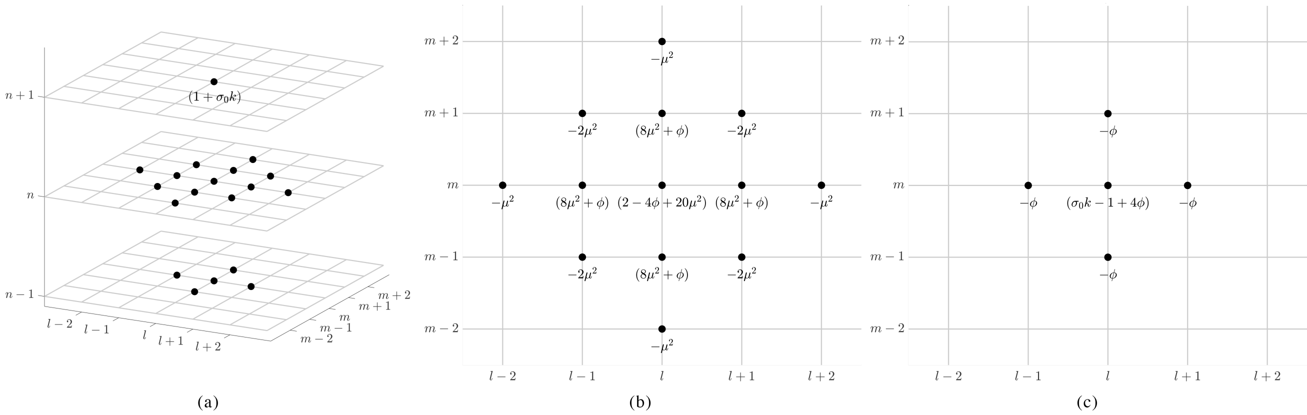

Figure 1. A visualisation of Equation (30). The dots and equations represent the locations (l, m) and what these are multiplied with. (a) An overview. (b) The current time-step n. (c) The previous time-step $n-1$.

5.2 Plate

$$
\begin{aligned}
(1+\sigma_0 k)u_{l,m}^{n+1} &= (2-4\phi+20\mu^2)u_{l,m}^n \\
&- \mu^2(u_{l,m+2}^n + u_{l,m-2}^n + u_{l+2,m}^n + u_{l-2,m}^n) \\
&- 2\mu^2(u_{l+1,m+1}^n + u_{l+1,m-1}^n + u_{l-1,m+1}^n + u_{l-1,m-1}^n) \\
&+ (8\mu^2+\phi)(u_{l,m+1}^n + u_{l,m-1}^n + u_{l+1,m}^n + u_{l-1,m}^n) \\
&+ (\sigma_0 k - 1 + 4\phi)u_{l,m}^{n-1} \\
&- \phi(u_{l,m+1}^{n-1} + u_{l,m-1}^{n-1} + u_{l+1,m}^{n-1} + u_{l-1,m}^{n-1}) \\
&+ k^2\delta_{l_\mathrm{e},m_\mathrm{e}}F_\mathrm{e},
\end{aligned}
\tag{30}
$$

where

$$\mu = \frac{\kappa k}{h^2}, \quad \phi = \frac{2\sigma_1 k}{h^2} \quad \text{and} \quad \delta_{l_\mathrm{e},m_\mathrm{e}} = \delta(x - x_{l_\mathrm{e}}, y - y_{m_\mathrm{e}}).$$

6. USER INTERACTION

User-controlled variables:

- Bowing position
- Bow force
- Bow velocity
- Connection points
- Finger position (pitch)

The vertical velocity of the finger is linked to the bow velocity with a maximum of $V_\mathrm{b} = 0.2$ m/s and the finger force is linked to the excitation function with a maximum of 100 m/s^2.

6.1 Sensel Morph

Something about the sensel morph

6.1.1 Mapping strategies

Something about the different prototype mappings, and the ”final” mapping

7. DISCUSSION

8. CONCLUSION AND FUTURE WORK

Acknowledgments

We would like to thank...

9. REFERENCES

[1] S. Bilbao, *Numerical Sound Synthesis, Finite Difference Schemes and Simulation in Musical Acoustics*. John Wiley and Sons, Ltd, 2009.

[2] ——, “A modular percussion synthesis environment,” *Proc. of the 12th Int. Conf. on Digital Audio Effects (DAFx-09)*, 2009.