FORUM ACUSTICUM EURONOISE 2025

A TWO-REGISTER HAPTIC INTERFACE FOR ARTICULATED CONTROL OF HARPSICHORD PHYSICAL MODELS

Matthew Hamilton* **Riccardo Russo** **Craig J. Webb**
Michele Ducceschi
NEMUS Lab, Dept. of Industrial Engineering, University of Bologna

ABSTRACT

A two-register haptic interface is presented for the articulated control of physically modelled harpsichord sounds. While physical modelling synthesis affords realism and nuance, standard MIDI-based controllers lack the resolution to capture the fine temporal detail required for historically informed keyboard performance. A custom three-key harpsichord mechanism was commissioned from an expert builder, and augmented with optical and force-sensitive sensors to track independent jack displacements per register. These data inform a real-time nonlinear string synthesis engine, enabling control over subtle inter-register plucking delays characteristic of historical instruments. The interface supports velocity-sensitive articulation across registers and integrates with a real-time software model employing modal energy-preserving techniques. The system reintroduces key expressive features absent from sample-based approaches, offering a viable solution for high-fidelity digital reconstruction of early keyboard gestures.

Keywords: *physical modelling, harpsichord, musical interface, MIDI controller, Virtual Studio Technology*

1. INTRODUCTION

The evolution of digital sound synthesis has transformed the relationship between human gesture, performance, and sound production. Early digital instruments offered sonic flexibility but introduced a degree of abstraction, primarily due to reliance on indirect parameter manipulation rather than direct physical interaction [1, 2]. With advances in computational power, real-time processing and expressive controllers have reintroduced immediacy to digital performance, allowing musicians to interact with sound more intuitively [3–5]. Nevertheless, the fundamental challenge of expressive control remains central across domains such as live electronic music and interactive performance systems, where continued exploration has led to the creation of complex multimodal interfaces [6].

The challenge of control remains open and becomes relevant when applied to historical musical instruments, where physical fragility often precludes direct interaction [7–9]. Physical modelling synthesis presents a promising solution by enabling real-time simulation of an instrument's behaviour [10, 11]. However, digital reconstructions often struggle to capture the fine-grained control and responsiveness required for historically informed performance practices.

The harpsichord exemplifies this difficulty. Characterised by its distinctive articulation and multiple registers, it requires nuanced control to produce its characteristic sound [12]. In traditional harpsichords, jack staggering creates intentional time offsets between multiple plucks triggered by the same key [13]. This prevents the player from engaging multiple strings with excessive force simultaneously. Yet, such subtle timing is often overlooked in sample-based libraries, where a single MIDI note triggers all registers simultaneously. Standard MIDI keyboards compound the issue, lacking the tactile feedback and mechanical complexity necessary to replicate historical instruments' expressive intricacies—especially the delayed plucks across registers. Previous work has

**Corresponding author*: matthew.hamilton2@unibo.it.
Copyright: *©2025 Matthew Hamilton et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.*

enhanced a museum-based keyboard replica with a sensor system [14] to control a commercial sample library, but sample-based implementations remain fundamentally limited. The core issue lies in their one-to-one mapping of MIDI events to sound triggers.

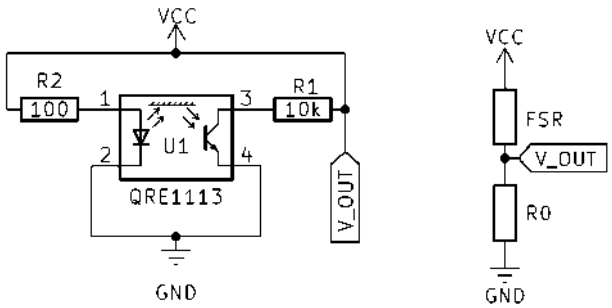
This paper introduces a novel control interface designed to overcome these constraints. By incorporating a custom sensing system that independently tracks the movement of each register, the proposed design enables precise articulation and timing control. This restores essential performance characteristics of historical harpsichords and strengthens the link between digital synthesis and authentic historical expression. A public repository containing supplementary material is available [1] . In particular, videos demonstrating the pluck detection mechanism are relevant for supporting the results discussed in the following section.

2. DETECTION MECHANISM AND INTERFACE PROTOTYPING

Rather than relying on commercial weighted-key MIDI controllers—which fail to replicate the mechanical resistance profile of a harpsichord [7]—a physical prototype of the harpsichord mechanism was commissioned from a professional instrument maker. The prototype includes three keys, each coupled to two jacks tuned in unison (Figure 1). While scaled down and unpitched, the string selection and mechanical dimensions were informed by a luthier to approximate the C2–D2 register with comparable tension and gauge. Each key acts as a class-1 lever, lifting jacks from different points along its length. This staggered design results in varying jack velocities, influencing the timing of plucks. Two complementary sensor types, QRE1113 reflective optical sensors and Interlink Electronics FSR 400 force-sensitive resistors (FRS), were employed to track gestures. The optical sensors—selected for their precision in short-range applications and prior success in musical interfaces—were directed at greyscale gradient tags attached to the sides of each jack. This setup largely draws from the original design by McPherson [15]. Mounted on custom PCBs (one per register), the sensors output voltage (`V_OUT`) in proportion to reflected infrared light, feeding analogue data into the microcontroller's ADC inputs (Figure 2(a)). Signal integrity was improved by shielding the open prototype construction against ambient light and power line interference through

[1] https://github.com/Nemus-Project/modal-harpsichord

Figure 1: Top and side views of the three-key model built to prototype harpsichord key mechanisms and sensing system.



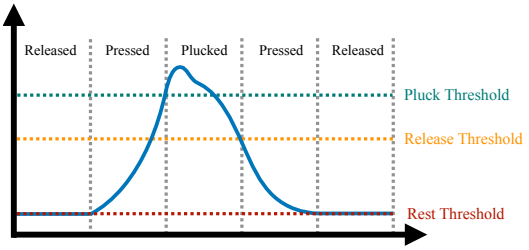
(a) QRE1113 (b) FSR

Figure 2: Sensor voltage divider circuits. In each case `V_OUT` connects to an ADC channel of the microcontroller.

both physical means (3D-printed cover plates) and software calibration (4-point moving average).

In a previous project involving the augmentation of a full-scale keyboard for the San Colombano Museum exhibition [14], a single-threshold detection system was implemented using only optical sensors. Although functional, it was prone to both false positives and missed events due to sensor noise, as occasional spikes led to unreliable threshold crossings during expressive play. Furthermore, artificially dragging the string with the plectrum, without producing an actual pluck, resulted in a series of false positives, as the detection system was unable to distinguish true plucks using simple thresholding alone.

A two-register pluck mechanism is reliable only if such false detections can be removed altogether. To




```
if(value > pluck_threshold AND state != PLUCKED)
   send_note_on(channel, note_number, velocity);
   state = PLUCKED;
else if(value < release_threshold AND state == PLUCKED)
   send_note_off(channel, note_number, velocity);
   state = PRESSED;
else if(value < rest_threshold)
   state = RELEASED;
```

Figure 3: Diagram of optical sensor signal transitioning through each state, and corresponding pseudocode.

achieve this, the current system uses a two-threshold hysteresis mechanism implemented as a finite state machine with three states: `RELEASED`, `PLUCKED`, and `PRESSED`. A MIDI `Note_ON` is triggered only when the sensor reading exceeds the `pluck_threshold` and the system is in the `RELEASED` state. Conversely, a `Note_OFF` is issued when the value falls below a lower `release_threshold`, and the system is in the `PLUCKED` state. This avoids instability from minor fluctuations near a single threshold and prevents rapid state toggling. Figure 3 presents a diagram of the hysteresis setup, and the corresponding pseudocode is given. Figure 4 presents snapshots of the plectrum-string interaction during a pluck, and the reflective sticker used by the optical sensing mechanism.

In the current design, FSR signals complement optical data by indicating actual force application. FSRs were installed beneath each jack to detect actual string excitation. Connected in a voltage divider circuit (Figure 2b), the output voltage reads:

$$V_{OUT} = \frac{R_0}{R_0 + R_{FSR}} V_{CC}, \tag{1}$$

from which the varying sensor's resistance R_{FSR} can be estimated. Such varying resistance can ultimately be converted to a force signal measured in Newton's via a conversion table from the supplier.

In addition to measuring force magnitudes, the FSR signal functions as a validation layer. When the optical

Figure 4: (Top): Snapshots of the plectrum-string interaction during a pluck. Note how the plectrum initially pushes the string but is eventually pushed back by the string before release. (Bottom): side view of a jack showcasing the colour gradient in the reflecting sticker used by the optical sensing mechanism.

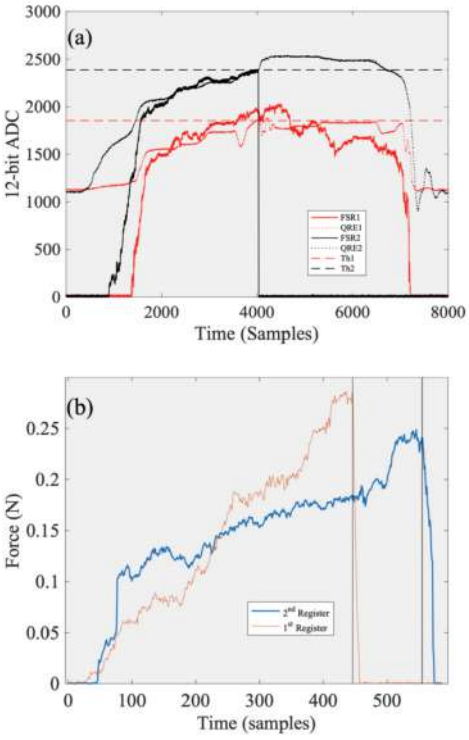


Figure 5: (a) FSR and optical sensor data for both registers of a key, as recorded by the ADC. **Th1** and **Th2** lines are the thresholds for the first and second set of optical sensors. (b) FSR data at 8 kHz sample rate. Black lines show the pluck point for each register. Time difference between plucks: 14 ms (110 samples).

sensor reading exceeds the pluck threshold, the system verifies whether a corresponding rate of change is present in the FSR signal. If no such drop is detected—indicating an absence of string contact—the event is disregarded. As a result, only deliberate plucking gestures lead to sound output, while unintended movements are effectively disregarded. Figure 5(a) displays the four signals captured from a single key gesture—two per register, representing the optical and force-sensitive sensors. It can be observed that one string is correctly identified as plucked, evidenced by a sharp drop in the force signal (black solid curve), while the second string (red curves), despite the optical threshold being crossed, is not plucked.

Figure 5(b) shows a time series from both sensors on a single key, sampled at 8 kHz and converted to Newtons. The onset of key motion is marked by displacement without resistance, followed by a sharp drop in the FSR signal that corresponds to the pluck. Together, the optical and force signals show strong temporal alignment across gestures. Based on recordings and sensor data, the delay between register plucks during normal performance ranged from 12ms to 30ms.

The current configuration supports per-register velocity assignment at note onset, with `Note_ON` velocities derived from the peak force measured by the FSR prior to its decay. Since FSRs are insensitive to key release, `Note_OFF` velocity is instead estimated from the rate of downward motion, as captured by the optical sensor when crossing the release threshold.

3. PHYSICAL MODELS AND IMPLEMENTATION

The control interface drives a nonlinear, two-register string model emulating the physical response of early Italian harpsichords. These instruments typically employed yellow brass strings, which, being relatively soft and unable to withstand high tension, contributed—together with the flexible soundboard and thin case sides—to a characteristically bright and sustained tone. For a digital reconstruction to sound convincing, it must account for behaviours like pitch glides, modal coupling, and amplitude modulation. Such features lie beyond the reach of linear string models and require the inclusion of geometric nonlinearities. Consider the following nonlinear model:

$$\rho A \partial_t^2 u(x,t) = \mathcal{L}u(x,t) + \partial_x \phi'_{nl}(\partial_x u) + \delta^{(x_f)} f(t), \quad (2)$$

where $u(x,t)$ denotes the transverse displacement; ρ is the material density; $A = \pi r^2$ is the cross-sectional area of a circular string; $\delta^{(x_f)} = \delta(x - x_f)$ is a Dirac delta representing the point of external excitation. The notation ∂_z^n denotes the n^{th} partial derivative along z, and primes ($'$) denote total differentiation. The operator $\mathcal{L}$ models linear dynamics:

$$\mathcal{L} := T_0 \partial_x^2 - EI \partial_x^4, \quad (3)$$

with T_0 the string tension, E Young's modulus, and $I = \pi r^4/4$ the area moment of inertia. The nonlinear potential function ϕ_{nl} governs the geometric nonlinearity:

$$\phi_{nl}(\partial_x u) := \frac{EA - T_0}{8} (\partial_x u)^4. \quad (4)$$

This expression approximates a low-order expansion of the geometrically exact string model (for the full model and a discretisation employing finite elements, see [16]), sufficient to produce perceptually relevant nonlinear effects such as pitch glides and modal coupling [17]. Note that the interactions with the soundboard and the radiation into the surrounding medium, whilst important for sound synthesis, are neglected at this stage, as they do not influence the structure of the control data stream of this proof-of-concept.

Boundary conditions are taken as simply supported: $u = \partial_x^2 u = 0$ at $x = \{0, L\}$. Applying modal decomposition:

$$u(x,t) := \sum_{m=1}^{M} X_m(x) q_m(t) = \mathbf{X}^\intercal(x)\, \mathbf{q}(t), \quad (5)$$

with $X_m := \sqrt{2/L} \sin\left(\frac{m\pi x}{L}\right)$ and $\mathbf{q}(t)$ the vector of modal coordinates, results in the following projected system:

$$\ddot{\mathbf{q}} = -\boldsymbol{\Omega}^2 \mathbf{q} - \mathbf{C}\dot{\mathbf{q}} - \int_{\mathcal{I}} \mathbf{X}' \phi'_{nl}\, \mathrm{d}x + \boldsymbol{\eta} f(t), \quad (6)$$

where $\boldsymbol{\Omega} := \mathrm{diag}[\Omega_1, ..., \Omega_M]$ is a diagonal matrix of eigenfrequencies defined as:

$$\Omega_m = \sqrt{\frac{T_0}{\rho A}\left(\frac{m\pi}{L}\right)^2 + \frac{EI}{\rho A}\left(\frac{m\pi}{L}\right)^4}. \quad (7)$$

The diagonal damping matrix $\mathbf{C} := \mathrm{diag}[2\sigma_1, ..., 2\sigma_M]$ was arbitrarily added after modal projection, and it approximates the light, uncoupled modal losses. The damping coefficients σ_m can be estimated experimentally or be derived from a model such as Cuesta and Vallette's [18].

3.1 Energy quadratisation and time stepping scheme

The modal system is passive and satisfies the energy balance:

$$\frac{\mathrm{d}H}{\mathrm{d}t} = -\dot{\mathbf{q}}^\intercal \mathbf{C}\dot{\mathbf{q}} + \dot{\mathbf{q}}^\intercal \boldsymbol{\eta} f(t), \tag{8}$$

with total energy:

$$H = \frac{1}{2}\dot{\mathbf{q}}^\intercal\dot{\mathbf{q}} + \frac{1}{2}\mathbf{q}^\intercal\boldsymbol{\Omega}^2\mathbf{q} + \int_{\mathcal{I}} \phi_{nl}\,\mathrm{d}x, \tag{9}$$

and note that the energy is strictly decreasing when the source is inactive ($f(t) = 0$). To ensure numerical stability in the presence of nonlinear terms, the system is reformulated using the Scalar Auxiliary Variable (SAV) method [19, 20]:

$$\psi := \sqrt{2\int_{\mathcal{I}} \phi_{nl}\,\mathrm{d}x}, \tag{10}$$

yielding:

$$\ddot{\mathbf{q}} = -\boldsymbol{\Omega}^2\mathbf{q} - \mathbf{C}\dot{\mathbf{q}} - \psi\nabla_{\mathbf{q}}\psi + \boldsymbol{\eta} f(t), \tag{11a}$$

$$\dot{\psi} = (\nabla_{\mathbf{q}}\psi)^\intercal\dot{\mathbf{q}}. \tag{11b}$$

System (11) is discretised using staggered time series for $\mathbf{q}$ and ψ. Let $\mathbf{q}^n \approx \mathbf{q}(Tn)$ and $\psi^{n-\frac{1}{2}} \approx \psi(T(n-\frac{1}{2}))$ at time step n and with sampling period T. Discrete-time operators are defined as:

$$\delta_2\mathbf{q}^n := \frac{\mathbf{q}^{n+1} - 2\mathbf{q}^n + \mathbf{q}^{n-1}}{T^2}, \tag{12a}$$

$$\mu_2\mathbf{q}^n := \frac{\mathbf{q}^{n+1} + 2\mathbf{q}^n + \mathbf{q}^{n-1}}{4}, \tag{12b}$$

$$\delta_\cdot\mathbf{q}^n := \frac{\mathbf{q}^{n+1} - \mathbf{q}^{n-1}}{2T}, \tag{12c}$$

$$\delta_+\psi^{n-\frac{1}{2}} := \frac{\psi^{n+\frac{1}{2}} - \psi^{n-\frac{1}{2}}}{T}, \tag{12d}$$

$$\mu_+\psi^{n-\frac{1}{2}} := \frac{\psi^{n+\frac{1}{2}} + \psi^{n-\frac{1}{2}}}{2}. \tag{12e}$$

Using these, the time-stepping scheme employed here is:

$$\delta_2\mathbf{q}^n = -\tilde{\boldsymbol{\Omega}}^2\mu_2\mathbf{q}^n - \tilde{\mathbf{C}}\delta_\cdot\mathbf{q}^n - \mu_+\psi^{n-\frac{1}{2}}\mathbf{g}^n + \boldsymbol{\eta} f^n,$$

$$\delta_+\psi^{n-\frac{1}{2}} = (\mathbf{g}^n)^\intercal\delta_\cdot\mathbf{q}^n,$$

$$\mathbf{g}^n := (\nabla_{\mathbf{q}}\psi)\big|_{t=Tn},$$

where:

$$\tilde{\Omega}_m^2 := \frac{4}{T^2}\frac{1 - 2e^{-\sigma_m T}\cos(T\sqrt{\Omega_m^2 - \sigma_m^2}) + e^{-2\sigma_m T}}{1 + 2e^{-\sigma_m T}\cos(T\sqrt{\Omega_m^2 - \sigma_m^2}) + e^{-2\sigma_m T}},$$

$$\tilde{\sigma}_m := \frac{4}{T}\frac{1 - e^{-\sigma_m T}}{1 + 2e^{-\sigma_m T}\cos(T\sqrt{\Omega_m^2 - \sigma_m^2}) + e^{-2\sigma_m T}}.$$

This discretisation has several desirable properties. First, it ensures that the linear part is discretised exactly [21], as it appears in previous works employing modal synthesis [20, 22]. Second, it is unconditionally stable. Finally, the system's update -not shown here for brevity- is performed explicitly via the Sherman-Morrison formula [17, 19, 23], avoiding nonlinear root finding algorithms -often requiring several iterations per time step- or large matrix inversions.

To manage computational complexity further, full nonlinearity is applied only to modes below 3 kHz; higher modes are treated linearly. This hybrid approach preserves perceptual features while reducing floating-point cost. Additionally, a constrained version of the SAV method, adapted from [20], is employed to enforce the non-negativity of $\mu_{t+}\psi^{n-\frac{1}{2}}$, preventing spurious numerical artefacts from appearing in the solution and allowing the auxiliary variable to decay toward zero. A detailed description can be found in [24, 25].

4. REAL-TIME SOFTWARE IMPLEMENTATION

The software component of the prototype system consists of a real-time instrument plug-in (VST3/Audio Unit), designed for integration with standard Digital Audio Workstations (DAWs). The selected DAW must support the transmission of MIDI channel information to the active track, enabling multi-channel polyphonic expression (MPE)-style communication.

4.1 System Overview

The software simulates a pair of nonlinear strings, forming a single voice whose pitch is determined by the mechanically actuated key. For the purposes of computational benchmarking, low-pitched notes were selected: specifically, C2 (65 Hz), along with the adjacent C#2 and D2. Each string is modelled with a length of 2 m and a radius of 0.3 mm, employing appropriate material parameters for brass.

The system processes `Note_ON` and `Note_OFF` events received over two independent MIDI channels, supporting individual excitation and damping of each

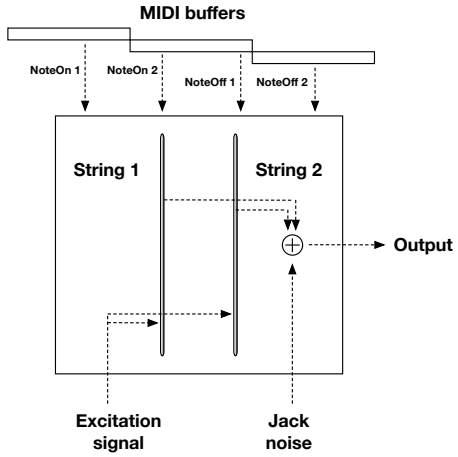


Figure 6: Real-time system implementation comprising a pair of nonlinear strings triggered by MIDI signals distributed across multiple buffers.

string and preserving temporal offsets between the associated jacks. These MIDI events may span multiple processing buffers; therefore, each is timestamped and rendered within the correct frame. Additional components include an excitation signal and jack noise, as illustrated in Figure 6.

Excitation is applied as a time-limited forcing term at a designated point along each string, informed by the pressure sensor beneath the mechanical key. The excitation signal, approximately 50 ms in duration, is selected via linear interpolation between two stored waveforms, based on the MIDI `Note_ON` velocity. Additionally, the sound of the jack returning to its rest position is incorporated via a pre-recorded noise sample. The `Note_OFF` velocity is used to modulate the amplitude of this signal prior to mixing with the output audio.

4.2 Computational Challenges

The use of nonlinear modal string synthesis in a real-time audio environment is computationally demanding, both at the model initialisation phase (`Note_ON`) and during continuous audio rendering at each time step. In contrast to a linear string model—which requires a single vector update and dot product per time step—the nonlinear system entails substantially more operations. For instance, the C2 note requires a modal resolution of 150 modes. Each time step involves two matrix-vector multiplications, seven vector updates, and six dot products. When matrices are square, multiplication accounts for over 80% of the total computational load.

To attain viable single-core CPU performance, both algorithmic and low-level code optimisations were implemented. As mentioned previously, one effective strategy involved partitioning the modal range into two segments: a low-to-mid frequency band up to approximately 3 kHz, and a higher frequency band extending to the upper modal limit. The lower segment (approximately 50 modes for C2) is computed using the full nonlinear model, whereas the higher modes are updated using a simplified linear routine. This segmentation reduces matrix dimensionality and lowers the cost of associated vector operations. With this approach, CPU usage per string was reduced to approximately 10%. Further optimisation is anticipated through under-sampling of the nonlinear modal segment.

Beyond real-time rendering, the initialisation of each string model introduces additional processing overhead. Specifically, the computation of modal shapes—involving thousands of trigonometric evaluations at each `Note_ON` event—can lead to transient CPU load spikes, potentially causing buffer under-runs. To mitigate this, a caching system for precomputed modal matrices is recommended for scalable polyphonic implementation.

5. DISCUSSION AND CONCLUSION

The system presented in this paper builds upon previous work on sensor-augmented harpsichord interfaces—most notably the San Colombano project [14]—by introducing substantial refinements in both gesture detection and synthesis integration. While earlier designs relied solely on optical displacement tracking, the present system introduces a dual-sensor architecture, combining reflective optical sensors with force-sensitive resistors (FSRs). This hybrid approach enables robust pluck detection through velocity-sensitive hysteresis and force validation, reducing false positives, particularly during complex or expressive gestural input. These developments address known shortcomings of threshold-based detection and allow for a more reliable mapping between mechanical gestures and sonic outcomes. The current prototype setup is visible in Figure 7.

Conventional digital control of plucked keyboard instruments suffers from several shortcomings, namely: the lack of control resolution in standard MIDI devices, the inability to articulate register-specific timing offsets, and the absence of mechanical feedback needed for histori-

Figure 7: The prototype setup, showing the three-key harpsichord model connected to a 2023 MacBook Pro. The prototype VST is visible, along with a list of triggered MIDI messages.

cally informed performance. The proposed system addresses these challenges through a custom-built harpsichord mechanism with staggered dual-register jacks, paired with real-time gesture tracking and nonlinear physical modelling. The resulting interface supports temporally resolved articulation across registers, required for expressive control.

From the perspective of sound synthesis, the system incorporates a physically realistic, nonlinear string model that supports two independently plucked registers per key. By constraining nonlinear computations to low-order modes (below 3 kHz), and employing a modal energy quadratisation scheme with a constrained Scalar Auxiliary Variable (SAV), the model achieves a balance between physical fidelity and computational tractability. As a result, real-time synthesis is attained on consumer-grade hardware with a CPU footprint of approximately 10% per voice on the most computationally demanding strings in the bass register, while preserving key nonlinear characteristics such as pitch glides and modal coupling.

The prototype shows that a high level of expressivity can be achieved through targeted hardware augmentation and physics-informed synthesis. It reaffirms the value of context-specific interfaces when coupled with physically grounded models, particularly in the domain of historically informed digital instrument design. Future work will extend the current architecture to a full keyboard, explore enhanced mapping strategies for velocity and dynamics, and incorporate more detailed models of soundboard interaction and acoustic radiation.

6. ACKNOWLEDGMENTS

This work was supported by the European Research Council (ERC) within the Horizon 2020 framework, with grant NEMUS-StG-950084.

7. REFERENCES

[1] C. Roads, *The Computer Music Tutorial*. MIT Press, 1996.

[2] F. R. Moore, *Elements of Computer Music*. Prentice Hall, 1990.

[3] S. Trolland, A. Ilsar, C. Frame, J. McCormack, and E. Wilson, "Airsticks 2.0: Instrument design for expressive gestural interaction," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (Auckland, New Zealand), pp. 16–21, 2022.

[4] M. Caren, R. Michon, and M. Wright, "The keywi: An expressive and accessible electronic wind instrument," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (Birmingham, United Kingdom), pp. 604–609, 2020.

[5] A. McPherson, "The space between the notes: Adding expressive pitch control to the piano," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (Daejeon, Republic of Korea), pp. 181–184, 2013.

[6] A. Tanaka and R. B. Knapp, "Multimodal interaction in music using the electromyogram and relative position sensing," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (Dublin, Ireland), pp. 171–176, 2002.

[7] K. B. McAlpine, "Sampling the past: A tactile approach to interactive musical instrument exhibits in the heritage sector," in *Innovation in Music 2013* (R. Hepworth-Sawyer, J. Hodgson, R. Toulson, and J. L. Paterson, eds.), KES Transactions on Innovation in Music, (York, United Kingdom), pp. 110–125, Future Technology Press, 2014.

[8] A. Baldwin, T. Hammer, E. Pechiulis, P. Williams, D. Overholt, and S. Serafin, "Tromba moderna: A digitally augmented medieval instrument," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (Brisbane, Australia), pp. 14–19, 2016.

[9] ERC, "NEMUS project." https://cordis.europa.eu/project/id/950084. Grant No. 950084, accessed 2025.

[10] J. O. Smith, "Physical modeling synthesis update," *Computer Music Journal*, vol. 34, no. 2, pp. 87–93, 2010.

[11] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. Chichester, UK: John Wiley & Sons, Ltd, 2009.

[12] E. L. Kottick, K. D. Marshall, and T. J. Hendrickson, "The acoustics of the harpsichord," *Scientific American*, vol. 264, no. 2, pp. 110–115, 1991.

[13] C. D. Veroli, "Optimising harpsichord staggering," *Harpsichord and Fortepiano*, vol. 16, pp. 8–13, 2012.

[14] M. Hamilton, M. Ducceschi, R. Livi, C. Vicens, and A. McPherson, "Augmentation of a historical harpsichord keyboard replica for haptic-enabled interaction in museum exhibitions," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, (Canberra, Australia), 2025. Under review.

[15] A. McPherson, "Portable measurement and mapping of continuous piano gesture," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 152–157, 2013.

[16] J. Chabassier and P. Joly, "Energy preserving schemes for nonlinear hamiltonian systems of wave equations: Application to the vibrating piano string," *Computer Methods in Applied Mechanics and Engineering*, vol. 199, no. 45-48, pp. 2779–2795, 2010.

[17] R. Russo, S. Bilbao, and M. Duccheschi, "Scalar auxiliary variable techniques for nonlinear transverse string vibration," in *Proceedings of the IFAC Workshop on Lagrangian and Hamiltonian Methods for Non Linear Control (IFAC)*, (Besançon, France), pp. 160–165, Apr. 2024.

[18] H. Cuesta and C. Vallette, "Nonlinear damping in musical string models," *Acta Acustica united with Acustica*, vol. 82, no. 4, pp. 657–664, 1996.

[19] S. Bilbao, M. Ducceschi, and F. Zama, "Explicit exactly energy-conserving methods for hamiltonian systems," *Journal of Computational Physics*, vol. 472, p. 111697, 2023.

[20] M. van Walstijn, V. Chatziioannou, and A. Bhanuprakash, "Implicit and explicit schemes for energy-stable simulation of string vibrations with collisions: Refinement, analysis, and comparison," *Journal of Sound and Vibration*, vol. 569, p. 117968, 2024.

[21] J. L. Cieśliński, "On the exact discretization of the classical harmonic oscillator equation," *Journal of Difference Equations and Applications*, vol. 17, no. 11, pp. 1673–1694, 2011.

[22] M. Van Walstijn, J. Bridges, and S. Mehes, "A real-time synthesis oriented tanpura model," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, (Brno, Czech Republic), pp. 175–182, 2016.

[23] J. Sherman and W. J. Morrison, "Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *Annals of Mathematical Statistics*, vol. 21, pp. 124–127, 1950.

[24] S. Bilbao, R. Russo, C. Webb, and M. Ducceschi, "Real-time guitar synthesis," in *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, (Guildford, UK), pp. 163–170, Sept. 2024.

[25] R. Russo, *Non-Iterative Numerical Simulation Techniques for Nonlinear String Vibration in Musical Acoustics*. Ph.d. dissertation, University of Bologna, Bologna, Italy, April 2025.