

TOWARDS A VIRTUAL-ACOUSTIC STRING INSTRUMENT

Sandor Mehes, Maarten van Walstijn, Paul Stapleton

Sonic Arts Research Centre

School of Electronics, Electrical Engineering and Computer Science

Queen's University Belfast

Belfast, UK

{smehes01,m.vanwalstijn,p.stapleton}@qub.ac.uk

ABSTRACT

In acoustic instruments, the controller and the sound producing system often are one and the same object. If virtual-acoustic instruments are to be designed to not only simulate the vibrational behaviour of a real-world counterpart but also to inherit much of its interface dynamics, it would make sense that the physical form of the controller is similar to that of the emulated instrument. The specific physical model configuration discussed here reconnects a (silent) string controller with a modal synthesis string resonator across the real and virtual domains by direct routing of excitation signals and model parameters. The excitation signals are estimated in their original force-like form via careful calibration of the sensor, making use of adaptive filtering techniques to design an appropriate inverse filter. In addition, the excitation position is estimated from sensors mounted under the legs of the bridges on either end of the prototype string controller. The proposed methodology is explained and exemplified with preliminary results obtained with a number of off-line experiments.

1. INTRODUCTION

Synthesis by physical modelling is designed as the ultimate methodology for digital simulation of real-world instruments [1–4]. The key difference with sample-based approaches is that the synthesis algorithm captures and parameterises the physical behaviour rather than the signal output. Hence in principle, *virtual-acoustic instruments* can be designed on this basis that are similar to real-world acoustical instruments in the way they sonically respond to player actions and afford performance nuances. However while significant advances have been made over the past few decades regarding numerical modelling of musical instruments, relatively little progress has been made so far in terms of real-time control of the resulting algorithms.

It is worthwhile noting at this point that the problem of synthesis control has been much more widely investigated as a *gestural mapping* problem (see, e.g. [5–8]). Generally this concerns a more free approach to the design of new, computed-based musical instruments, usually with-

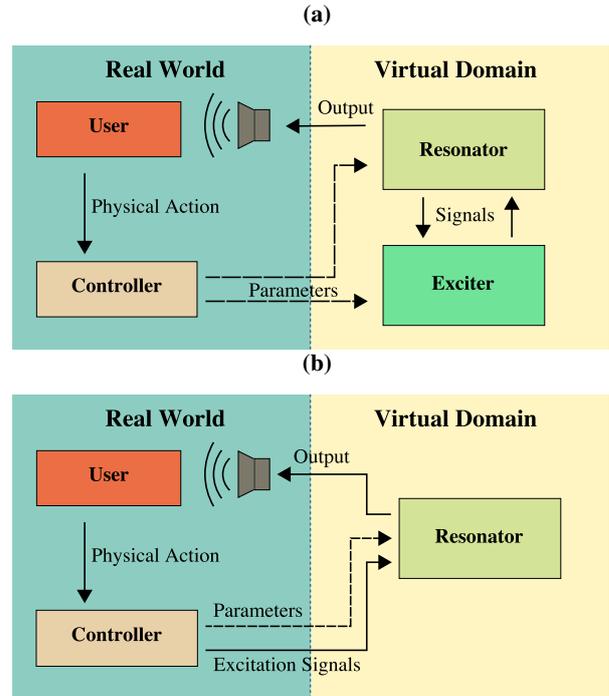


Figure 1: (a) A ‘conventional’ configuration for real-time control of a physical model. (b) An alternative configuration in which the excitation signals are generated with the controller.

out specific consideration of the constraints and characteristics of acoustics instruments, and often instilling a more loose coupling between the player and the instrument. The mapping problem does in fact not exist in the same way when using physical modelling synthesis, since the model parameters generally have direct counterparts in the real world. This suggests that the mapping can in principle be replaced by a *direct routing* between a real-world controller and a virtual-domain sounding system (i.e. the physical model algorithm), with the interface dynamics directly inherited from the modelling process [9]. Fig. 1a illustrates this concept schematically. Following several earlier studies, the physical model is represented here in terms of its block decomposition into an “exciter” (i.e. an excitation object such as a bow or a finger) and a “resonator” (i.e. a vibrating structure such as a string or a membrane). Traditionally, the exciter, the resonator, and their interaction are all modelled (i.e. existing in the virtual domain), with as-

sociated parameters that are to be controlled by the player. For example, playing a bowed string model involves real-time adjustment of the bow parameters (e.g. bow speed, bow force) as well as of the string parameters (e.g. finger stopping position). One of the main challenges in realising such a conventional physical model configuration arises from the high computational costs involved in precise modelling of all of the physical mechanisms involved (see, for example, the case of a two-polarisation bow-string model [10]).

Leaving aside such efficiency concerns, the remaining challenge focuses on controller design, which invariably involves perpending the larger scope of multi-modal interaction, i.e. also including forms of haptic and visual feedback. This topic has been extensively investigated in the past few decades within the sound and computing community as well as in the wider realm of human-computer interaction, and has resulted in various strands of related controller design concepts, including those based on *natural* [11], *tangible* [12, 13], *embodied* [12], *enactive* [13] and *effortful* [14] interaction. The current paper is partly inspired by these concepts, and in alignment with them seeks a sensor configuration that minimises its interference with the instrumentalist’s actions.

In light of such interaction design criteria, Berdahl and Smith [15] proposed a slightly different configuration for physical model control, which leaves the exciter part in the real-world domain (see Fig. 1b). In this arrangement, the physical form of the controller resembles the main vibrating element of the simulated instrument. In the case studied in [15], the player is presented with a (silent) controller interface with two strings, one of which is damped and excited in the usual ways (plucking, striking, etc.) in order to drive a physics-based string resonator model, while the other controls the pitch.

A key technical challenge that arises in this approach is to ensure that the interface dynamics are captured in appropriate form for driving the virtual-domain sound resonator, which boils down to ‘clean extraction’ of the relevant excitation signal(s). That is, the controller should comprise real-time sensing/processing of signals in order to obtain an equivalent of the signal(s) normally flowing from the exciter to the resonator. For example, in the case of percussive string excitation, the signal that is most suited to excite a virtual string model is the actual force signal exerted by the player on a (strongly damped) string, with any possible distortion by the setup (e.g. coloration by the sensors) removed as much as is feasible. In addition, the envisaged application to performance requires a high-quality audio, low-noise excitation signal. In [15] this is addressed by using an electric guitar as the tangible interface, fitted with undersaddle piezoelectric pickups to sense the string vibrations. The piezos are more suitable than bridge pickups due to the inherently nonlinear characteristics of the latter. The (augmented) use of the electric guitar as the physical controller has consequences regarding the type of control and idiom a performer is invited to engage with. The probability of players and virtual-acoustic instrument designers reinventing string playing in a way that genuinely expands

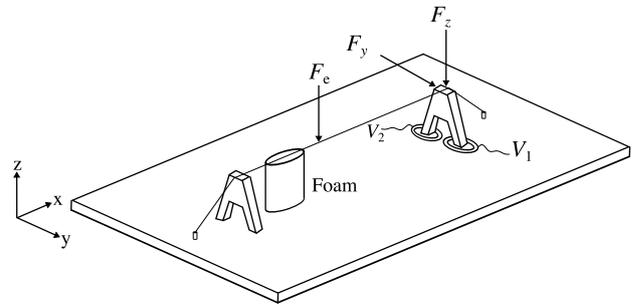


Figure 2: Setup of the prototype string controller. Under each leg of the bridges (a, b) piezoelectric disks are placed, generating voltages (V_1 , V_2) resulting from the pressure of the vibrating string on the bridges.

artistic practices is further reduced if the string model parameters are restricted to a range that produces sounds that are close to the sonic palette of conventional guitar sounds.

This paper is similarly motivated, but takes a different approach by moving away from using the electrical guitar and associated commercial piezo pickups for the prototype string controller. The main purpose is to keep more design flexibility, which aligns with the longer-term aim of more adventurously exploring the acoustic affordances of virtual-acoustic string instruments. The principal technical novelty of the work presented here is that specific attention is given to removing the characteristics of the sensor via linear filtering and pre-calibration. In addition, a method for estimating the excitation position (from the same sensor data) is proposed. The excitation type is restricted largely to percussive styles (i.e. resulting from short interactions between the string and a finger/object), since the technical challenges involved in separating the ‘exciter’ from the ‘resonator’ are considerably more complex for fully sustained excitation (i.e. string bowing).

The remainder of the paper is structured as follows. The prototype string controller is presented in Section 2, including the signal processing used for estimating the player force signal and the excitation position from the vibrations sensed at the instrument bridge. Section 3 then gives a summary of the modal synthesis string resonator model and its implementation, followed by the exposition of a few exemplifying preliminary off-line results in Section 4.

2. A PROTOTYPE STRING CONTROLLER

2.1 Experimental Setup

A string is stretched over two bridges mounted on a wooden support platform, as depicted in Figure 2. Currently the bridges of a Guzheng (a Chinese string instrument [16]) are used. A piece of foam is placed close to the left bridge with the purpose of damping the vibrations of the string, as such subduing multiple round-trip wave reflections. Assuming linear wave propagation and neglecting string stiffness and damping this means that - apart from at very low frequencies - any transversal force seen at the bridge furthest from the foam is approximately equal to a delayed version of the force wave travelling towards bridge generated when

the player excites the string

$$F(t) \approx F_e \left(t - \frac{L - x_e}{c} \right), \quad (1)$$

where $L = 0.460\text{m}$ is the string length between the two bridges, x_e is the excitation position, and c is the velocity at which transversal waves travel along the string. Hence the foam placement allows directly extracting the excitation force from $F(t)$, be it with latency $\tau = (L - x_e)/c$.

The setup also features foam strips that are glued to the bottom of the support platform to absorb external vibrations that could corrupt the signal. To sense the vibrations at the bridge, a piezoelectric disk (PD) is positioned under each of its two legs. The analogue signal routing for these sensors contains a high-pass circuit which helps attenuating the DC component (including any signal ‘drift’ that would be detrimental to any further processing). The piezo sensor signals are digitally captured with an NI USB-6215 data acquisition platform. The final stage of the processing chain is a computer¹ for both the parameter estimation and the real-time implementation of the string resonator model.

2.2 Excitation Force Signal Estimation

Referring again to Fig. 2, the strategy here is to set up an inferential sensing system to estimate the vertical (F_z) and horizontal (F_y) forces exerted on the bridge by the string. Under the assumption of linear behaviour of both the bridge and the sensors, the PD signals — denoted here as $V_i(t)$, where $i = 1, 2$ — are simply filtered versions of the force signals. Under vertical force excitation, the frequency-domain relationships then are:

$$V_i(\omega) = G_i(z)F_z(\omega), \quad (2)$$

where $G_i(z)$ is the corresponding transfer function, encapsulating the characteristics of the bridge, $F_z(\omega)$ is Fourier transform the vertical force excitation on the bridge and $V_i(\omega)$ is the Fourier transform of the piezo signal. In order to estimate the vertical force signal, the relationships in eq (2) must be inverted. Fig. 3 illustrates how this can be realised in the time domain, using calibration filters $C_i(z)$ that approximate the inverses of the transfer functions $G_i(z)$ (see the next Section for the design of these filters).

For vertical forcing of the bridge, the string pushes downwards on the two legs simultaneously, resulting in in-phase piezo signals. Therefore summing the filtered signals and multiplying by $1/2$ gives an estimation of the vertical force component by averaging; this is realised with the upper arm of the signal processing diagram in Fig. 3. On the other hand, with a horizontal force impact on the bridge one leg of the bridge is lifted up while the other leg is pushed down resulting in signals that are out of phase with each other. Therefore a horizontal force estimate \hat{F}_y can be obtained by subtracting the filtered signals from the PD. The difference in exciting in the vertical as opposed to the horizontal plane can in itself be considered as a filtering

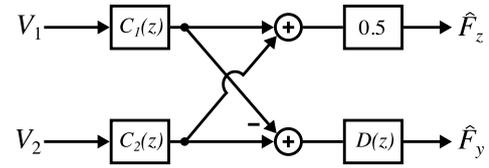


Figure 3: Signal diagram for the force estimation. The voltages (V_1, V_2) are generated by the piezoelectric disks processed by the calibration filters, $C_1(z)$ and $C_2(z)$. The sum and difference give estimation of the bridge forces (\hat{F}_z, \hat{F}_y). Filter $D(z)$ can be added for fine tuning the \hat{F}_y estimation.

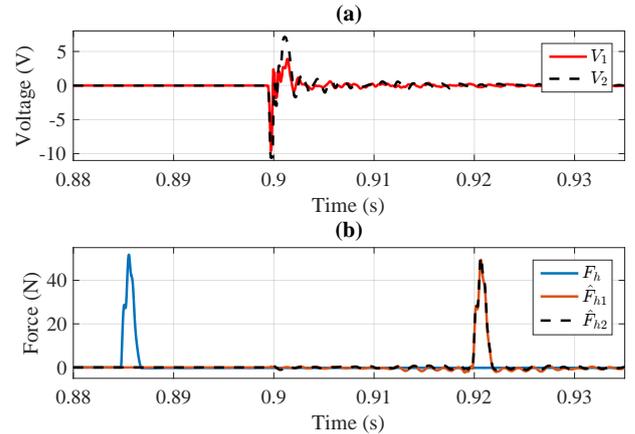


Figure 4: Voltages (a) generated by piezoelectric disks placed under the bridge. Force measured with an impact hammer (b) applied at the bridge (blue) and estimations from V_1 (red) and V_2 (black, dashed).

effect, and can thus potentially be modelled with an additional filter $D(z)$ for increased accuracy for \hat{F}_y ; this extension has not been yet realised or tested within the project though, and instead the difference signal is currently taken directly as a measure of the vertical bridge force.

2.3 Identification of the Calibration Filters

In order to estimate the forces exerted on the bridge from the sensed PD signals as described above, digital filters $C_i(z)$ that approximate the inverses of $G_i(z)$ are required. To obtain such filters, a pre-calibration experiment is carried out by measuring the force impact on the bridge when it is hit by an impact hammer² from above, and simultaneously sensing the PD signals. An example set of measurement signals is plotted in Figure 4. An optimum inverse filter can then be designed for each piezo through various means; here adaptive filtering methods [17] are applied to provide a first, preliminary result, in the form of an FIR filter. In particular the recursive least squares (RLS) algorithm is useful in this case because of its relatively high robustness against input signal characteristics. A suitable set of input/output training signals is created by first convolving the hammer and piezo signals with a white pseudo-noise signal, ensuring that the input signals are of sufficient

¹ iMac 2.8GHz quad-core Intel Core i5, 16 GB of 1867MHz LPDDR3 onboard memory

² Dytran Dynapulse 5800B4

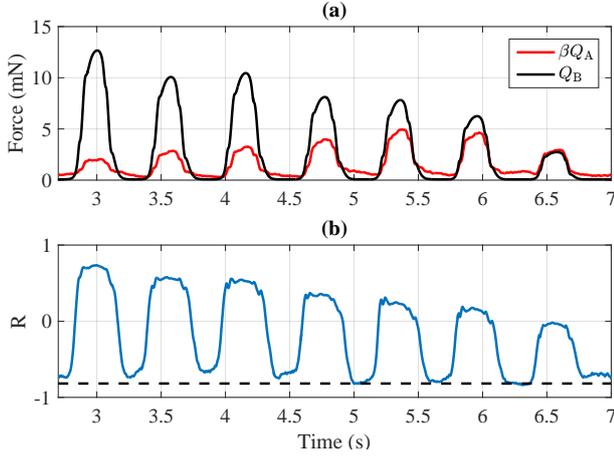


Figure 5: Position estimation signals obtained when striking the string successively at positions $x_e = 373.5, 353.0, 332.5, 312.0, 291.5, 271.0,$ and 250.5mm , respectively. (a) Short-time rectified average of the two estimated force signals. (b) The corresponding evolution of the variable $R(t)$. The dashed line indicates the ‘default value’ $(1 - \beta)/(1 + \beta)$ which it approximately attains in the absence of excitation.

length to train the RLS algorithm. The error signal (the difference between the target and estimate signal) is defined allowing for a small time delay of the FIR filter’s impulse response; the results presented used at a sampling rate of 51.2kHz . As seen in Figure 4(b), filtering $V_i(t)$ through the calibration filters this way results in an accurate estimation of $F_z(t)$ from each PD (and thus also from the averaged signal obtained with the upper arm of the signal diagram in Figure 3).

Note that the equalisation that is carried out by passing the piezo signal through the calibration filter affects both amplitude and phase characteristics. As a result, sharp force pulses are reconstructed by the calibration filters from piezo signals that are more ‘smeared’ over time. This means that transient-rich detail in the excitation signal (arising from the player’s interaction with the string) is exposed more sharply in the final audio signal than if the piezo signal were to be passed straight to the resonator model.

A drawback of the adaptive filtering approach is that using long FIR filters can be computationally demanding, making it less suitable for real-time application. A more efficient approach is possible though, by first extracting the dominant modes of the bridge and implementing these separately as second-order resonance filters [18].

2.4 Excitation Position Estimation

In order to estimate the position at which the string is excited by the player, piezo sensors are also fitted under the legs of the other bridge. One approach would be to determine the time difference between the signals arriving at the bridges, with pulses due to plucks positioned closer to the right-hand bridge (B) arriving earlier at that bridge than at the left-hand side bridge (A). However the presence of the foam, which causes temporal smearing of wave pulses

travelling towards bridge (A) complicates this approach. Instead the estimation approach taken here is based on determining the short-time RMS-like signal averages $Q_A(t)$ and $Q_B(t)$ at the two bridges. For each bridge, this signal is calculated by first applying a first-order low-pass filter to the estimated force, then applying signal rectification (by taking the absolute value the signal), and finally applying a smoothing (moving average) filter. Fig. 5(a) shows an example set of signals when the string is struck successively at seven different positions. The two signals are then used in the calculation of the dimensionless quantity

$$R(t) = \frac{Q_B(t) - \beta Q_A(t)}{Q_B(t) + \beta Q_A(t)}, \quad (3)$$

where β is a constant compensating for the foam damping (here $\beta = 10$ is used); the damping by the foam is approximately constant within the low-frequency band that is effectively used in the signal calculation. Fig. 5(b) shows how $R(t)$ varies with striking position. In periods of no excitation, the value of R is approximately $(1 - \beta)/(1 + \beta)$ due to the noise on the signals from which $Q_A(t)$ and $Q_B(t)$ are calculated. More generally, $R(t)$ relates to $x_e(t)$ through a static nonlinear mapping $R = \mathcal{G}(x_e)$; this map can be obtained by a further pre-calibration measurement involving multiple plucks at a range of positions along the string followed by a curve fitting. Fig. 6 shows an example result of this process, in which we retrieved the map in its simplest possible form, i.e. a straight line. This procedure prepares for the estimation in real-time of the excitation positions in the range $[L/2 - L]$ employing the inverse of the obtained mapping, i.e.

$$x_e(t) = \mathcal{G}^{-1}(R(t)). \quad (4)$$

The inconsistencies in R seen in Fig. 6 for any of the string excitation positions are due to (a) extraneous measurement signals and (b) string vibrations resulting from effectively stopping the string with the plucking object (hence setting up wave roundtrips over the string portion between $x = x_e$ and $x = L$). The latter problem is unavoidable to certain extent, but the former can be alleviated by improved signal conditioning.

3. STRING RESONATOR MODEL

3.1 String Model

Transversal string vibrations, taking into account non-idealities such as stiffness and damping, can be described with the partial differential equation [3, 4]

$$\rho A \frac{\partial^2 y}{\partial t^2} = T \frac{\partial^2 y}{\partial x^2} - EI \frac{\partial^4 y}{\partial x^4} - \gamma(\beta) \frac{\partial y}{\partial t} + \mathcal{F}_e(x, t), \quad (5)$$

in which ρ , A , T , E , and I are mass density, cross-sectional area, tension, Young’s modulus, and moment of inertia, respectively, and where $y(x, t)$ denotes the transversal displacement at axial position x and time t . Given that the support platform is thick, strong, and heavy, simply supported boundary conditions can be assumed:

$$y(x, t) \Big|_{x=0, L} = 0, \quad \frac{\partial^2 y}{\partial x^2} \Big|_{x=0, L} = 0. \quad (6)$$

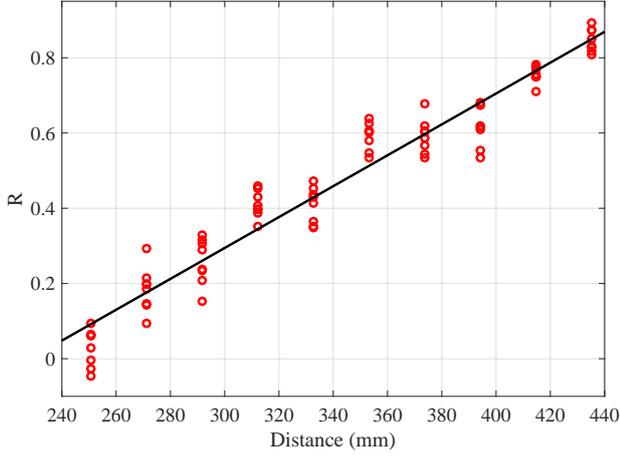


Figure 6: Mapping between R and x_e . The circles indicate individual measurement data points, and the line is a least-squares fit to the data.

Frequency-dependent string damping is incorporated here by defining the parameter $\gamma(\beta)$ in (5) as:

$$\gamma(\beta) = 2\rho A \left[\sigma_0 + (\sigma_1 + \sigma_3 \beta^2) |\beta| \right], \quad (7)$$

where β is the wave number and $\sigma_{0,1,3}$ are physically-motivated fit parameters. The external excitation is restricted here to a single point, i.e.

$$\mathcal{F}_e(x, t) = \delta(x_e) F_e(t), \quad (8)$$

where $F_e(t)$ and x_e represent the force signal and excitation position, both of which are directly obtained from the controller within the proposed approach. An appropriate audio signal can be obtained by calculating the termination force at $x = L$:

$$F_T(t) = -T \frac{\partial y}{\partial x} \Big|_{x=L} + EI \frac{\partial^3 y}{\partial x^3} \Big|_{x=L}, \quad (9)$$

and inputting this to a body filter, such as those designed in [18]. Alternatively, the string velocity at a pick-up position x_p can serve as a sound output signal.

3.2 Modal Synthesis

The solution of (5) can be expressed as a superposition of the normal modes of the string (indexed with i):

$$y(x, t) = \sum_{i=1}^M v_i(x) \bar{y}_i(t), \quad (10)$$

where $\bar{y}_i(t)$ denotes the mode displacement and $v_i(x) = \sin(\beta_i x)$ is the corresponding mode shape (spatial eigenfunction) for the boundary conditions given in (6), with $\beta_i = i\pi/L$. Substitution of (10) into (5), then multiplying with $v_i(x)$ and applying a spatial integral over the length of the string yields that the dynamics of each of the modes is governed by:

$$m \frac{\partial^2 \bar{y}_i}{\partial t^2} = -k_i \bar{y}_i(t) - r_i \frac{\partial \bar{y}_i}{\partial t} + v_i(x_e) F_e(t), \quad (11)$$

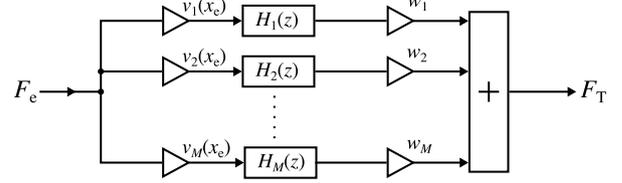


Figure 7: Signal diagram for the modal synthesis algorithm. The output weights w_i are computed with (13) or alternatively set as $v_i(x_p)$ for velocity pickup at $x = x_p$.

where $k_i = \frac{1}{2} L (EI \beta_i^4 + T \beta_i^2)$ and $r_i = \frac{1}{2} L \gamma(\beta_i)$ are the elastic and damping constants of the mode, respectively. The term $m = \frac{1}{2} \rho A L$ is the modal mass, which is the same for all modes. Under the assumption of each mode being *under-damped* (i.e. $r_i < 2\sqrt{k_i m}$), the modal frequencies are $\omega_i = \sqrt{k_i/m - \alpha_i^2}$, where (in accordance with (7))

$$\alpha_i = r_i/(2m) = \sigma_0 + \sigma_1 \beta_i + \sigma_3 \beta_i^3 \quad (12)$$

are the modal decay rates. The modal expansion of the termination force is

$$F_T(t) = \sum_{i=1}^M \underbrace{\left[-T v_i'(L) + EI v_i'''(L) \right]}_{w_i} \bar{y}_i(t), \quad (13)$$

where $v_i'(x)$ and $v_i'''(x)$ denote the first and third spatial derivative of $v_i(x)$, respectively.

The dynamics of each of the modes can be simulated in discrete time by discretising (11), for example using the impulse-invariant method [19], which exactly preserves the modal parameters ω_i, α_i . Denoting the transfer functions of the resulting digital model oscillators with $H_i(z)$, a modal synthesis structure that implements equation (10) then takes the form as illustrated in Figure 7; this modal sound synthesis engine structure is essentially the same as those proposed in various earlier studies (see, e.g. [3, 20]).

Two instances of this processing structure are created in order to simulate vibrations in two polarisations; this allows emulating beating effects due to a slight difference in effective length between the y - and z -polarisations.

3.3 Real-Time Parameter Control

An early real-time prototype has been implemented in Max MSP³ using the *resonators*-⁴ object for the realisation of 1024 modal oscillators. The *resonators*- parameters are calculated with a dedicated external that translates MIDI controlled string parameters into modal parameters. This external utilises a frequency envelope function in order to avoid rendering aliased modes or clicks when varying parameters that affect the mode frequencies, ensuring that modes smoothly fade out when nearing the Nyquist frequency and fading in when the mode frequency falls below Nyquist.

³ <https://cycling74.com/products/max/>

⁴ http://cnmat.berkeley.edu/files/maxdl/archive/CNMAT_Externals-MacOSX-1.0-78-gd490ddd.tgz

4. PRELIMINARY EXPERIMENTS

In order to get a glimpse of what a virtual-acoustic string instrument of the proposed design might sound like, various explorative experiments were carried out. Piezo signals were recorded during a session in which a player applied forces to the string using various exciters, including a finger, plectrum, and a bow. These signals were processed off-line using the calibration filters in order to obtain estimations of the applied force signals, which were in turn fed to the modal resonator synthesis engine described in Section 3. For comparison, the modal resonator was also driven directly with the piezo signals, which yields sounds having the ‘nasal’ timbre typically associated with piezo-resistive disks. This effect is significantly reduced by the calibration filters. Sound examples can be found on the accompanying webpage⁵. Further off-line exploration focused on using ‘out of range’ geometrical string parameters (length, cross-section), which allows exposing inherent string properties such as stiffness on a different time scale.

5. CONCLUSIONS

Physical models have been developed and implemented in real-time for several decades now. A rare example of turning a physical model into an exciting new virtual-acoustic instrument is the Kalichord [21], which departs from the configuration discussed in this paper in that it incorporates physical controller features of kalimbas and accordions in its design. The off-line results presented here are intended to give an initial impression of the wider possibilities of virtual-acoustic string instruments if specific attention is given to controller design that attempts to capture the interface dynamics in the form of an estimated excitation signal.

Some promising initial results are obtained, but several improvements are needed to more fully achieve the intended aims. Firstly, the signal conditioning needs to be improved in order to meet the signal-to-noise ratio requirements for this type of application. Secondly, in order to develop the potential of the approach more fully, the design needs to be targeted to more specific instruments, probably using extended models with well-tuned parameters. Finally, the next versions of the string controller will have to be more robust and ergonomic for application in performance.

A further consideration for future exploration is of a more abstract nature. The motivation behind playing a virtual rather than a real resonator stems from the fully parameterised nature of the virtual, i.e. one is free to change any of the physical parameters, thus having an instrument that houses a broad family of a certain type rather than one fixed instantiation. As discussed in [22], this concept can be extended to on-line variation of physical parameters that are not normally accessible in real-world instrument. Thus, once the virtual-acoustic instrument is functioning well in the sense of emulating both the acoustic and interface dynamics, an adventurous next step would be to explore extended control by real-time adjustment of a wider range of the available physical parameters.

6. REFERENCES

- [1] V. Välimäki, J. Pakarinen, C. Erkut, and M. Karjalainen, “Discrete-time modelling of musical instruments,” *Reports on Progress in Physics*, vol. 69, no. 1, pp. 1–78, Jan. 2006.
- [2] J. O. Smith III, *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects*. W3K Publishing, 2010.
- [3] L. Trautmann and R. Rabenstein, *Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method*. Kluwer Academic/Plenum Publishers, New York, 2003.
- [4] S. Bilbao, *Numerical Sound Synthesis: Finite Difference Schemes and Simulation in Musical Acoustics*. Chichester, UK: John Wiley and Sons, 2009.
- [5] J. Bowers and S. O. Hellström, “Simple Interfaces to Complex Sound in Improvised Music,” in *CHI '00 Extended Abstracts on Human Factors in Computing Systems*, ser. CHI EA '00. New York, NY, USA: ACM, 2000, pp. 125–126.
- [6] A. Hunt and R. Kirk, “Mapping strategies for musical performance,” *Trends in Gestural Control of Music*, vol. 21, pp. 231–258, 2000.
- [7] T. Mudd, S. Holland, P. Mulholland, and N. Dalton, “Investigating the effects of introducing nonlinear dynamical processes into digital musical interfaces,” in *Proceedings of Sound and Music Computing Conference, 2015*. Sound and Music Computing Network, Jul. 2015.
- [8] M. M. Wanderley, “Gestural control of music,” in *International Workshop Human Supervision and Control in Engineering and Music*, 2001, pp. 632–644.
- [9] D. Menzies, “Composing instrument control dynamics,” *Organised Sound*, vol. 7, no. 3, pp. 255–266, Dec. 2002.
- [10] C. Desvages and S. Bilbao, “Two-Polarisation Finite Difference Model of Bowed Strings with Nonlinear Contact and Friction Forces,” in *Proc. of the 18th Int. Conference on Digital Audio Effects*, Trondheim, Norway, Dec. 2015.
- [11] S. Mann, “Natural Interfaces for Musical Expression: Physiphones and a physics-based organology,” in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 118–123.
- [12] E. Hornecker, “A Design Theme for Tangible Interaction: Embodied Facilitation,” in *ECSCW*, vol. 5. Springer, 2005, pp. 23–43.
- [13] S. O’Modhrain and G. Essl, “PebbleBox and Crumble-Bag: tactile interfaces for granular synthesis,” in *Proceedings of the 2004 conference on New interfaces for musical expression*. National University of Singapore, 2004, pp. 74–79.

⁵ www.socasites.qub.ac.uk/mvanwalstijn/smcl6/

- [14] P. Bennett, N. Ward, S. O'Modhrain, and P. Rebelo, "DAMPER: a platform for effortful interface development," in *Proceedings of the 7th international conference on New interfaces for musical expression*. ACM, 2007, pp. 273–276.
- [15] E. Berdahl and J. O. Smith III, "A Tangible Virtual Vibrating String," in *Proceedings of the 2008 Conference on New Interfaces for Musical Expression (NIME08)*, 2008.
- [16] C. Zheng and Y. Knobloch, "A Discussion of the History of the Gu zheng," *Asian Music*, vol. 14, no. 2, p. 1, 1983.
- [17] S. Haykin, *Adaptive Filter Theory*, 5th ed. Upper Saddle River, N.J.: Pearson Education, Jun. 2013.
- [18] M. Karjalainen and J. Smith, "Body Modeling Techniques for String Instrument Synthesis," in *CCRMA Papers Presented at the 1996 International Computer Music Conference Hong Kong*. Hong Kong, China: CCRMA, 1996, pp. 35–42.
- [19] J. G. Proakis and D. G. Manolakis, "IIR Filter Design by Impulse Invariance," in *Digital Signal Processing*, 2nd ed. New York, USA: Macmillan Publishing Company, 1992, pp. 623–627.
- [20] F. Avanzini, R. Marogna, and B. Bank, "Efficient synthesis of tension modulation in strings and membranes based on energy estimation," *The Journal of the Acoustical Society of America*, vol. 131, no. 1, pp. 897–906, 2012.
- [21] D. Schlessinger and J. O. Smith, "The Kalichord: A Physically Modeled Electro-Acoustic Plucked String Instrument." in *NIME*. Citeseer, 2009, pp. 98–101.
- [22] S. Orr and M. van Walstijn, "Modal representation of the resonant body within a finite difference framework for simulation of string instruments," in *12th International Conference on Digital Audio Effects, Como, Italy*, 2009.