



AALBORG UNIVERSITY
DENMARK

Aalborg Universitet

Perceptual Relevance of Haptic Feedback during Virtual Plucking, Bowing and Rubbing of Physically-Based Musical Resonators

Onofrei, Marius George; Fontana, Federico; Serafin, Stefania

Published in:
Arts

DOI (link to publication from Publisher):
[10.3390/arts12040144](https://doi.org/10.3390/arts12040144)

Creative Commons License
CC BY 4.0

Publication date:
2023

Document Version
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

Citation for published version (APA):

Onofrei, M. G., Fontana, F., & Serafin, S. (2023). Perceptual Relevance of Haptic Feedback during Virtual Plucking, Bowing and Rubbing of Physically-Based Musical Resonators. *Arts*, 12(4).
<https://doi.org/10.3390/arts12040144>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal -

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Article

Perceptual Relevance of Haptic Feedback during Virtual Plucking, Bowing and Rubbing of Physically-Based Musical Resonators

Marius George Onofrei ^{1,†} , Federico Fontana ^{2,*,†}  and Stefania Serafin ¹ ¹ Multisensory Experience Lab, CREATE, Aalborg University, 9220 Aalborg, Denmark² Human Computer Interaction Lab, University of Udine, 33100 Udine, Italy

* Correspondence: federico.fontana@uniud.it; Tel.: +39-0432-558432

† These authors contributed equally to this work.

Abstract: The physics-based design and realization of a digital musical interface asks for the modeling and implementation of the contact-point interaction with the performer. Musical instruments always include a resonator that converts the input energy into sound, meanwhile feeding part of it back to the performer through the same point. Specifically during plucking or bowing interactions, musicians receive a handful of information from the force feedback and vibrations coming from the contact points. This paper focuses on the design and realization of digital music interfaces realizing two physical interactions along with a musically unconventional one, rubbing, rarely encountered in assimilable forms across the centuries on a few instruments. Therefore, it aims to highlight the significance of haptic rendering in improving quality during a musical experience as opposed to interfaces provided with a passive contact point. Current challenges are posed by the specific requirements of the haptic device, as well as the computational effort needed for realizing such interactions without occurrence during the performance of typical digital artifacts such as latency and model instability. Both are however seemingly transitory due to the constant evolution of computer systems for virtual reality and the progressive popularization of haptic interfaces in the sonic interaction design community. In summary, our results speak in favor of adopting nowadays haptic technologies as an essential component for digital musical interfaces affording point-wise contact interactions in the personal performance space.

Keywords: physical modeling synthesis; musical haptics; vibrotactile feedback; musical interfaces



Citation: Onofrei, Marius George, Federico Fontana, and Stefania Serafin. 2023. Perceptual Relevance of Haptic Feedback during Virtual Plucking, Bowing and Rubbing of Physically-Based Musical Resonators. *Arts* 12: 144. <https://doi.org/10.3390/arts12040144>

Academic Editors: Justin Paterson and Marcelo M. Wanderley

Received: 8 January 2023

Revised: 12 February 2023

Accepted: 20 February 2023

Published: 7 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Nowadays, several musical instrument computer interfaces include haptic feedback as part of their output (Pietrzak and Wanderley 2020). The idea is not new (Berdahl et al. 2007), as testified by several decades of development of, e.g., the Cordis–Anima framework (Leonard and Cadoz 2015). In fact, it found fertile ground in considerations (Luciani et al. 2009) about the interaction with traditional musical instruments, in which the exchange of energy arising from the human instrumental gesture and the haptic feedback at the interaction point creates a physical coupling between musician and instrument. This back-and-forth energy coupling supports the notion that “an instrument [...] becomes an extension of the body”, an impression shared by many musicians (Nijs et al. 2009, p. 7) which could enforce the musical experience and open up flows and ideas also concerning musical improvisation.

The relatively recent availability of affordable force and especially vibratory devices has shed light on the effects that haptics have during performance in digital instruments otherwise passive to touch (Miranda and Wanderley 2006). Such effects have been shown to play a significant role, for instance, on the perception of quality in violins (Wollman et al. 2014) and pianos (Fontana et al. 2017), and the research field has gained

enough relevance to become known as musical haptics (Papetti and Saitis 2018). While vibrations are relatively simple to reproduce in a digital music instrument communicating with a host computer, kinematic cues require more elaborate actuation. In fact, the former propagate from the source to touchable points along the physical interface and, for this reason, can be delivered by one or more vibrotactile transducers placed somewhere on its body. Conversely, the latter are rendered by generating position-dependent force vectors reproducing static as well as moving 3D object's surfaces. Physical object simulation is a non-trivial task especially if the virtual object is moving, and normally requires the use of a robotic device. Technologies aiming at such simulations without the help of a robot, though appealing, are not mature enough yet to precisely reproduce the properties of a solid object (Georgiou et al. 2022; Hwang et al. 2017). On the other hand, the same devices set constraints in terms of electric power, encumbrance, portability, and mechanical noise during operation that often get worse their accuracy in rendering force increases. Together, such constraints can make the haptic prototyping of a digital music instrument unreasonable, if not impossible. Not surprisingly, vibrotactile augmentation has been introduced in digital pianos (Guizzo 2010) and prototype mixers (Merchel et al. 2010) with relatively low technical effort, whereas force feedback was able to find a place in the musical instrument market only in products affording specific interactions, mainly rotation (Behringer 2021; Native Instruments 2021).

In general, it is rare that force feedback technologies have the possibility to render nuances that are perceptually relevant for the skilled performer. Beyond some main macroscopic effects which are pleasantly experienced by every practitioner, haptic rendering in fact is not necessarily appreciated by skilled groups of users. Hence, haptic components in a digital music instrument multisensory design project should be introduced only when they determine a significant leap in interaction quality. The risk otherwise is of ending up with an inaccurate interface, furthermore exposing the aforementioned constraints and limitations. In this regard, point-wise interaction with one or more musical resonators represents an affordable design case. During this interaction, a force actuator renders only one point, with clear advantages in terms of device mechanical complication (Mihelj and Podobnik 2012) and effort for computing the force feedback at runtime; moreover, the same interaction gains musical significance if the resonator starts to oscillate during and after contact: this dynamic behavior implies that the contact model incorporates a damped mass-spring system (Rocchesso et al. 2003). Existence of a spring oscillating at audio frequency in the contact model means that a robotic device is not required to simulate extremely stiff contacts, thus featuring moderate reaction force with its actuators. Based on this premise, such actuators can also be engineered to produce low or no electromechanical noise during operation. Nor does the computation of the same model usually ask for high computing power. In other words, the haptics of point-wise contact sounds in many cases allow for use of less encumbering, sometimes even portable, inexpensive devices hosted by a standard computer. Ultimately, this means that musical haptics is, if not simple, technologically feasible in the case of plucking or bowing.

Within this set of technical and hardware constraints, the sonic interaction designer is asked to inject creative solutions in the digital instrument project. Only on a first approximation do these solutions target the resources at hand to the realization of a conventional artistic goal. Digital musical instruments based on simulations of physical interactions in fact have the potential to reach more ambitious goals than the mere reproduction of traditional instruments, since their simulation is not unconditionally constrained by physics. Their elements' acoustic properties and interconnections, for instance, can be dynamically varied over time, allowing for sounds throughout a performance that transcend their real-world equivalents while being still structurally tied to them. Moreover, the physics-based modeling approach to synthesis allows for more flexible control than, for example, an approach based on sound samples. Consider, for example, the case of a virtual violin, where the player controls the position of the bow along the string, as well as the force and velocity of bowing; all these performance parameters are physical and hence can be

incorporated in the model. Conversely, the latter approach would ask for recording of all possible sounds, combining the corresponding parametric conditions.

This merging between technical constraints and artistic intention represents the most exciting challenge for a sonic interaction designer. Frequently, such a challenge must step back in light of insurmountable issues. However, not so rarely, some such issues are circumvented by substituting unreachable interaction goals with workarounds that can be still appealing and, sometimes, especially interesting. Specifically concerning musical haptics, physics-based models are ideal candidates for enabling multimodal interactions, as the multisensory outputs of the interface are all computed from the same model. Favorable situations, moreover, exist when a musical haptic feature that was believed to be impossible to realize unexpectedly emerges instead as an inherent by-product of the physical interface. Usually such features go beyond the designer's knowledge of the technology at hand, and often they are uncovered later during early testing, while exploring a prototype behavior. We will see examples of these situations while describing our prototypes.

Content of This Paper

With these considerations in mind, the use of audio-haptic interfaces for controlling digital musical instruments based on physical models of interactions between exciters and musical resonators has a great deal of potential, as the possibility of a coherent multimodal interaction can improve the quality of the musical experience. Therefore, the current authors have recently investigated a number of such projects aiming to highlight the perceptual relevance of haptic feedback during physical interactions with musical resonators, focusing on the instrumental gestures of plucking and bowing and also on rubbing as an unusual opportunity for musical interaction.

Plucking relies on a quite simple mechanics, especially if acted through a plectrum. For this reason, it found a place in middle-age instruments such as the harpsichord and clavichord (Rossing 2010). In this case plucking consists of stretching a string by pulling it in one point until it is released and, hence, oscillates and generates sound. If the subtleties of this mechanics are neglected, such as those taking place when a string is plucked by a human finger, then a simple physical model can be realized which puts emphasis not on the dynamic aspects of plucking, but rather on the flexibility of the software implementation and consequent ease of applicability of external libraries. This way, a rich multisensory virtual scenario can be realized with moderate software development efforts. Following this design approach, an application was developed, among others, by the current authors within a popular programming environment for computer games and virtual scenarios and then validated with guitar players.

Friction is a common excitation mechanism for musical resonators, with bowed string instruments being probably the best known within the family of frictional musical instruments. Other subcategories of such instruments are friction drums and friction idiophones. Their force feedback during playing assists the player in dynamically controlling the instrument toward a target sound. Harsher, noisier sounds are connected to a more jagged response, whereas more harmonic sounds provide smoother feedback. Two projects exploring the frictional haptic interaction with virtual physically-based resonators were recently carried out by the current authors. Rubbing was realized on different physical interfaces using the same damped mass-spring model, with the goal of defining an increasingly tight interaction between the performer and instrument, but with a unique musical gesture and model underneath. Bowing expands on this simple physical model and investigates a similar audio-haptic interaction with virtual elastic strings, providing realistic haptic rendering of friction during bowing.

Our contribution corroborates two hypotheses, both aligned along a commonly recognized position in sonic interaction design:

- The inclusion of currently available programmable force feedback devices brings significant benefit to virtual plucking, bowing, and rubbing in terms of realism, hence

increasing the general quality of a point-wise interaction with a digital string independently of the specific physics-based model realizing it;

- The extension of the control space consequence of this inclusion offers, in connection with physics-based modeling, opportunities for interaction designers and artists to expand, also mutually, beyond known point-wise interactions with a virtual string.

After a synthetic introduction in Section 2 about the features a haptic device should have to fit a plucked, bowed, or rubbed digital instrument, an overview of our projects and their outcomes is given, respectively, in Sections 3–5, along with references to the publications in which they were initially published in part or extensively. This overview is followed by a short discussion about the potential merit of these projects and their interest for the reader of this journal.

2. Haptic Feedback Device for Point-Wise String Interaction

In spite of their different nature, plucking and bowing share characteristics that make it possible to generate interactive force feedback using the same haptic device for both such gestures. In fact, both can be mediated by an object that can be grasped using two fingers (respectively, the plectrum and the bow), and both establish a point-wise contact with the same resonator, that is, one musical instrument string (or more than one in the case of bowing multiple strings). In other words, any haptic device exposing a robotic arm that can be grasped using two fingers and actuates force vectors compatible with those reacted from one string (or more) in principle fits plucked, as well as bowed, digital music interface designs. An exception exists if plucking is acted directly using the finger: in this case, there is no bowing counterpart of finger-based plucking, and there is no guarantee that both musical gestures can be implemented using the same haptic device.

Accessibility to a digital music interface affording point-wise haptic feedback becomes reasonable for a performer if the haptic device provides some desirable features: (i) a lightweight, unencumbering, graspable robotic arm, (ii) an interaction point ranging within a circle parallel to the string-board by approximately 20 cm around a central point, (iii) force feedback compatible with the elastic reaction of a musical instrument string, (iv) low noise. Furthermore, sufficient portability and robustness make the device more attractive, along with moderate cost and energy consumption.

The 3DSystems Touch and Touch X robotic arms, shown in Figure 1, respond to such requirements.



Figure 1. 3D Systems Touch (left) and Touch X (right) haptic device.

The motors of the first device exert limited force (up to 3.3 N) and can actuate rotation and lift. On the other hand, its arm movements are extremely precise (tracking accuracy equal to about 0.055 mm at 1 kHz sampling frequency), silent, and prompt, as the technology has been designed to simulate materials that are not too stiff. The Touch X features characteristics that make it approximately twice as strong and accurate compared to the Touch, however, it is built on a heavier body. A complete list of such features can be found at <https://www.3dsystems.com/haptics-devices/touch> (accessed 22 February 2023).

In addition to responding with realistic forces, the motors of these two devices are able to follow little changes in the contact point position. In this way they can render friction and, hence, bowing and string rubbing during plucking. In other words, if a violinist bows the virtual string or a guitarist shifts the plucking point as when rubbing the plectrum along a wound string, then they will have the tactile feeling of scraping the string depending on its coating, with a net effect on the perceived realism of the interaction.

Parallel to feedback, the Touch and Touch X have good characteristics, such as input interfaces for capturing plucked and bowed interactions. Thanks to their spatial and temporal sampling accuracy, these devices in fact can capture the plectrum position with, e.g., a 1 mm accuracy when the performer's hand plucks or bows the string at 10 m/s, which is far above reported hand strumming velocities, normally below 1 m/s (Romero-Ángeles et al. 2019), as well as bow speed, which usually never goes above 30 cm/s (Guettler et al. 2003).

Finally, our candidate devices can of course be applied to contact interactions that are unconventional in the musical performance domain. This is the case of rubbing a spatially-distributed resonator, as it will be described in the following of this paper.

3. Plucking

Plucking is by its own nature a fully haptic interaction. When plucking a string, a musician typically uses her fingers or a plectrum to pull the string away from the body of the instrument and then release it. At that moment the string starts to vibrate and make sound. While the pitch of the sound is mainly driven by the string's physical parameters, a musician can achieve subtle control of the sound timbre by calibrating the force and velocity of the excitation, making plucking a versatile technique that can produce a wide variety of sounds, from percussion-like and rhythmic to smooth and sustained. Plucked instruments include lead and bass guitars, mandolins, ukuleles, harps, and dulcimers.

The keytar is a virtual electric guitar affording point-wise physical plucking with a plectrum, and keyboard-based selection of multiple notes. In addition to audio, the interface includes a screen displaying the guitar under control, and a touch device providing somatosensory feedback. This application has been previously documented concerning its initial implementation (Passalenti and Fontana 2018), subsequent hardware/software refinement through the addition of keyboard-based note control (Fontana et al. 2019), and further reworking of the string model (Fontana et al. 2020) leading to the final prototype which is reported here, mainly based on content taken from the last reference. A software build and/or code of Keytar is available upon request to the Authors.

As Figure 2 shows, Keytar's users pluck a string through the robotic arm using their dominant hand meanwhile selecting notes or chords through the keyboard controller with the other hand. As the dominant hand feels the resistance and textural properties of the string actuated by the robot, the plucking force follows by a perception-and-action process which comes natural for most musicians, especially guitarists. When a string is plucked, its vibration is also shown on the screen along with the action of the plectrum.

The keytar follows music interaction design projects based on physical interfaces with different realism and resemblances to stringed instruments, the most famous of which is Guitar Hero (Miller 2009). Previous plucked instrument models have been prototyped also in the form of virtual reality interfaces, sound synthesis models, or educational scenarios. Virtual Air Guitar is an ungrounded personal setting controllable through graspable accelerometers and camera hand tracking (Karjalainen et al. 2006). Similarly, Virtual Slide Guitar (Pakarinen et al. 2008) relies on high-rate infrared visual capture, later generalized into a programming framework (Figueiredo et al. 2009) targeting "air" control devices such as the Kinect (Hsu et al. 2013). Other plucked virtual instruments include the harp (Taylor et al. 2007) and kalichord (Schlessinger and Smith 2009). Augmentation of guitar playing has also been used for learning (Liarokapis 2005) and rehabilitation through music (Gorman et al. 2007).



Figure 2. Keytar interface.

The keytar was developed using the Unity3D environment, a now popular visual programming environment for prototyping computer games and virtual reality scenarios. Its flexibility in generating accurate application code ignited attention in virtual instrument developers, later suggesting (Manzo and Manzo 2014; Zhaparov and Assanov 2014) and now actively promoting (<https://rapsodos.ru/watch/How-to-Create-a-Piano-in-Unity-3D/>, accessed 22 February 2023) the use of this programming environment. Developers who are not in the digital instrument design business who however choose to populate a virtual scenario with musical instruments can rely on specific *assets* developed for a specific purpose (<https://assetstore.unity.com/packages/3d/props/musical-instruments-pack-20066>, accessed 22 February 2023).

Having a closer look at how the keytar was developed, and how it finally performs, offers a useful lesson about the pros and cons of relatively fast digital instrument prototyping such as that provided by Unity3D and similar environments. Readers will appreciate that Unity3D offers a fairly rich set of design tools to interaction designers that can be rapidly learned. However, it is rare that the same tools are able to address fine-grained features of plucking and bowing unless the developer knows where to code at low levels on the existing software architecture. In this case, such features can be more efficiently addressed by developing within environments such as JUCE—see Section 4.3.

Instead of squeezing the Unity3D assets beyond their limits, their creative use (especially concerning the visual scenario, together with a careful assessment of their rendering effects on the haptic feedback and overall multimodal simultaneity and coherence of the interface) allowed the keytar to perform surprisingly well, as testified also by the evaluations that will be shown at the end of this section.

3.1. Haptic Feedback

Unity3D makes available a *haptic workspace*, in which the material properties of all objects populating a scene can be specified thanks to the support provided by the Openhaptics plugin for Unity3D. Since, in particular, an object can be set to be invisible, this workspace concept ideally allows for defining extremely rich haptic scenarios. Unfortunately, complex workspaces often encumber the simulations, causing occasional crashes of the application particularly on slower computers especially if enabling contacts between objects which are not primitive in Unity3D. These crashes follow exceptions raised by the collision detection procedures that are continuously invoked by a software application in charge of rendering the physical contact between virtual objects. In fact, the computational resources such procedures can employ to enable the haptic workspace must be conciliated with the

requirements coming from the other (e.g., visual and audio) rendering engines. As a result, the tracked position of the robotic arm and its virtual counterpart computed by the collision detection algorithm during a virtual contact can mismatch to an unrecoverable level.

Even in absence of a system crash, the plectrum–string contact can go unnoticed by the system due to the real-time constraint imposed to the collision detector for returning a result. This is a well known problem in game design, sometimes referred to by Unity3D developers as the *bullet through paper problem*. (<https://answers.unity.com/questions/176953/thin-objects-can-fall-through-the-ground.html>, accessed 22 February 2023) Finally, unnoticeable spatiotemporal incongruities in, e.g., a computer action game, can be conversely perceived during more accurate, typically musical instrument interactions implemented using Unity3D.

These incongruities were least reduced by providing the haptic workspace with as few objects as possible. The plectrum visible in Figure 3 was enabled with an active contact region limited to the tip, by instantiating an invisible *sphere* object having a minimum (i.e., almost point-wise) diameter. In parallel, each haptic string was made of a chain of *cylinder* objects, active both in the haptic and visual workspace. Both spheres and cylinders are primitive in Unity3D and expose parameters of diameter, position, and length that could be set, along with compatible haptic properties. As opposed to the bullet through paper problem, a comparable “plectrum through string problem” leads to worse consequences than just seeing the plectrum diving into the fretboard. In fact, a plectrum after plucking must be eventually moved back. To avoid that a string stops the plectrum on its way back after being mistakenly passed through, an invisible haptic surface was added immediately below the strings, opposing maximum stiffness against penetration. The inclusion of this additional object overall resulted in the haptic workspace shown in Figure 3.

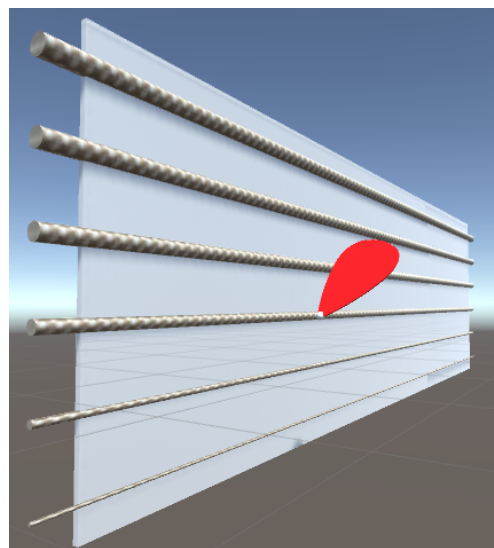


Figure 3. Haptic workspace in the keytar. The colliding sphere is visible as a white tip on the edge of the plectrum.

A subtle visuohaptic mismatch manifested when the haptic edge of the plectrum touched, but did not penetrate a string enough. In this case the haptic objects did interact, yet they did not collide deeply enough to elicit sufficient force feedback to the user. This problem was solved by wrapping each string with an invisible but touchable layer, which was set to be as thick as the radius of the haptic sphere forming the plectrum. The layer is shown in Figure 4. It ensured that users, while plucking the visible string, were moving the stylus on a position providing robust interpenetration between the plectrum and the string.

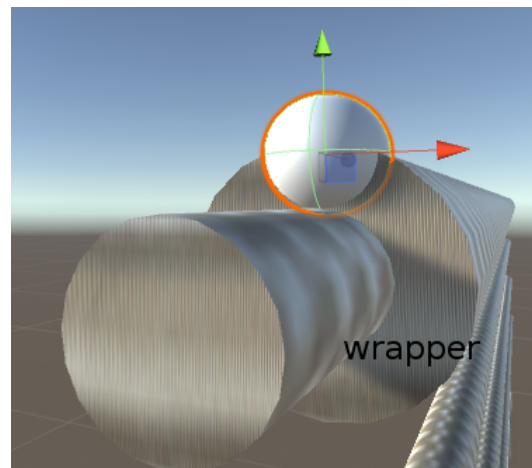


Figure 4. Mismatching positions of visual and haptic collision point. Wrapping with a touchable layer.

Such a haptic workspace models most of the physical interactions that occur during plucking. Essentially, two aspects, however, were not dealt with: one is the string elasticity; the other concerns the frictional effects arising during plectrum–string contact. Surprisingly, both were featured at no design cost by the touch device. In fact, as mentioned in Section 2, the device motors are not strong enough to simulate stiff objects, however they can still render elastic contacts and hence are suitable for reproducing a musical string. In parallel, the same motors are able to capture fine-grained surface properties. Hence, it was sufficient to alternate two different sizes of the small cylinders forming the string. The resulting tiny discontinuities between adjacent cylinders are accurately rendered when the stylus of the device is slid along a string, with an evident scraping effect of the plectrum. This vibrotactile effect could be even changed by varying the properties across different strings, in terms of their size (proportional to the diameter of the cylinders) and texture (proportional to the length and diameter differences). Thanks to these features coming directly for free from the actuation technology, strings could be defined as static objects in the sense that their position did not change in time, as opposed to the plectrum.

3.2. Visual Feedback

Visual feedback was designed by placing our two dynamic components, plectrum and strings, over a passive background depicting a guitar in a simple rehearsal room. As opposed to the plectrum, whose movement is directly manipulated by the performer, the strings needed to be animated by following plucking with visual patterns of vibration. These patterns were enabled by the Ultimate Rope Editor asset (<https://assetstore.unity.com/packages/tools/physics/ultimate-rope-editor-7279>, accessed 22 February 2023) available from the Unity3D Marketplace. This asset allows one to compose visual strings made of a variable number of sections and nodes. Then, several pseudorealistic physical parameters such as string tension, string weight, and breaking force can be set on each string, determining its dynamic behavior in response to an external force.

Due to its flexibility and dynamic processing of string vibrations, this asset provides excellent visual accuracy. The relationship with the haptic workspace is shown in Figure 5 along with the forces acting on each vibrating section as a consequence of plucking.

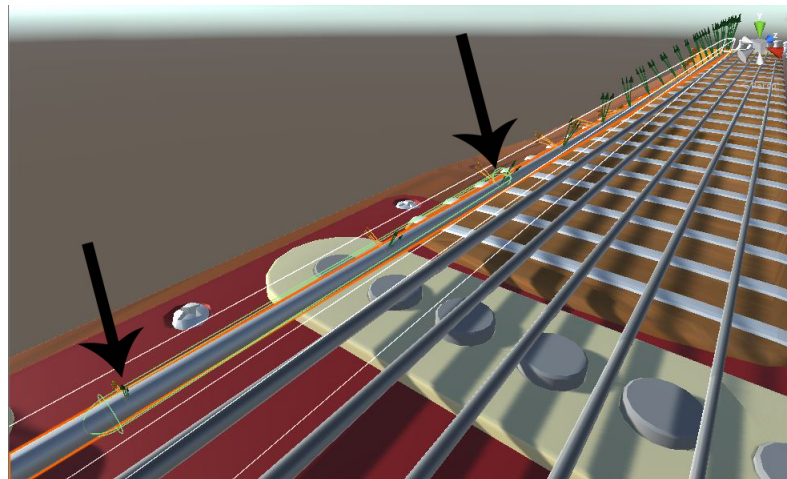


Figure 5. Aspects of the visual feedback in the keytar enabled by Ultimate Rope Editor. The active region of collision is surrounded by green circles surrounding the first string, each indicated by a black arrow. Small green and red arrows depict horizontal and vertical components of tangential force vectors exerted by the vibrating string sections.

At this point, it would seem natural to integrate the visual and haptic workspaces through the features provided by this asset. As Ultimate Rope Editor online computes the dynamics of each section on a string, why not export the data to the latter workspace? This attempt proved unsuccessful. First, strings whose linear density and tension were set to parameters compatible with the guitar become unstable, causing occasional chaotic oscillations of some sections. On the other hand, choosing parameter values fitting at least partially the physics of the guitar still provides satisfactory visual string vibration, yet with inconsistent haptic feedback. Both issues, which persisted also after reducing the locus of interaction to the neighborhood of the pick-up region, were ultimately caused by the excessive effort demanded by Ultimate Rope Editor to compute the dynamics of musical strings, along with an unsuitable frame rate for this asset of the Openhaptics collision detection algorithm, again causing the plectrum to occasionally move beyond a string instead of plucking it. Newer versions of Unity3D allow one to detect collisions through a callback procedure; this feature could alleviate such shortcomings in the keytar. In general, matching the requirements of the visual and haptic workspace when they collaborate together to form a virtual reality scenario is not straightforward, and significantly depends on the nature and characteristics of the interactive objects within it.

3.3. Auditory Feedback

A real-time procedural which models physical-based string sounds at a low computational cost needed to be selected. The Karplus–Strong (Karplus and Strong 1983) sound synthesis algorithm was chosen for this purpose, available as an asset (<https://github.com/mrmikejones/KarplusStrong>, accessed 22 February 2023) in Unity3D. This algorithm must be initialized in the sound intensity and tone fundamental frequency every time a string is plucked. The former parameter was set based on the spatial coordinates (x, y, z) of the contact point immediately before (b) and after (a) the contact was available at runtime from the haptic workspace. The distance between such two coordinates in fact provides a measure of the initial oscillation amplitude of a string immediately after plucking. By dividing the distance over time, accessible frame by frame through the static variable *Time.deltaTime*, an estimation of the velocity was found for initializing the sound intensity parameter of the algorithm:

$$\text{velocity} = \frac{\sqrt{(x_a - x_b)^2 + (y_a - y_b)^2 + (z_a - z_b)^2}}{\text{Time.deltaTime}}$$

The tone fundamental frequency was then set by reading data from the keyboard in Figure 2, thanks to Keijiro Takahashi's Musical Instrument Digital Interface (MIDI) communication software for Unity3D. (<https://github.com/keijiro/MidiJack>, accessed 22 February 2023)

Low latency and moderate computational costs helped us generate effective interactive musical feedback. On the other hand, Karplus–Strong by definition synthesizes string oscillations whose sound is far from that of a guitar. Unity3D provides an excellent collection of post-processing audio effects that can be cascaded to form multiple *AudioMixer* objects. Such objects were applied to the synthesized sounds, one for each string. The keytar adopts a traditional “Less to More” electric guitar pedalboard configuration including distortion, flanging, chorusing, compression, second distortion, and finally parametric equalization, all available as built-in effects in Unity3D. Such effects were parameterized for each string using subjective criteria, such as consistency among string sounds and similarity to a specific guitar tone.

The auditory feedback included real sounds coming from the robotic arm while actuating the string discontinuities colliding with the plucking plectrum. As a direct by-product of the haptic feedback, these sounds greatly improve the realism of the scenario. In fact, the alternating diameter of the cylinders visible in Figure 3 reproduced a texture similar to that of a wound string coating. Hence, if for example the plectrum is shifted longitudinally along a string, the motors actuate a motion similar to friction producing not only haptic, but also acoustic cues of convincing quality and sensory coherence.

3.4. Evaluation

The multisensory feedback conveyed by the keytar has already been judged to be especially satisfactory during showcasing to expert musical interface developers at the DAFx2019 conference on digital audio effects (Fontana et al. 2019) and furthermore in controlled experimental settings (Passalenti et al. 2019). Concerning the latter, in a virtual reality experiment, twenty-nine participants with years of regular practice on musical (including stringed) instruments were asked to first pluck the strings of a real guitar, and then to wear an Oculus Rift CV1 head mounted display showing an electric guitar and a plectrum in a nondescript virtual room. On top of the keytar, the contact detection engine controlled also a vibrotactile actuator standing below the touch device, hence producing additional vibrations independent of the force feedback. A within-subjects study compared four randomly balanced haptic conditions during plucking: no feedback (N), force only (F), vibration only (V), and force and vibration together (FV).

Each participant evaluated five attributes on a Likert scale (see plots labeled (a)–(e) in Figure 6). At the end of the test, each participant was additionally asked to choose the preferred condition. Finally (see the unlabeled plot (bottom right) in Figure 6) all the errors made by plucking the wrong string instead of a visually marked string during a part of the test involving individual plucking were logged. Results suggest the existence of significant effects of haptic feedback on the perceived realism of the strings.

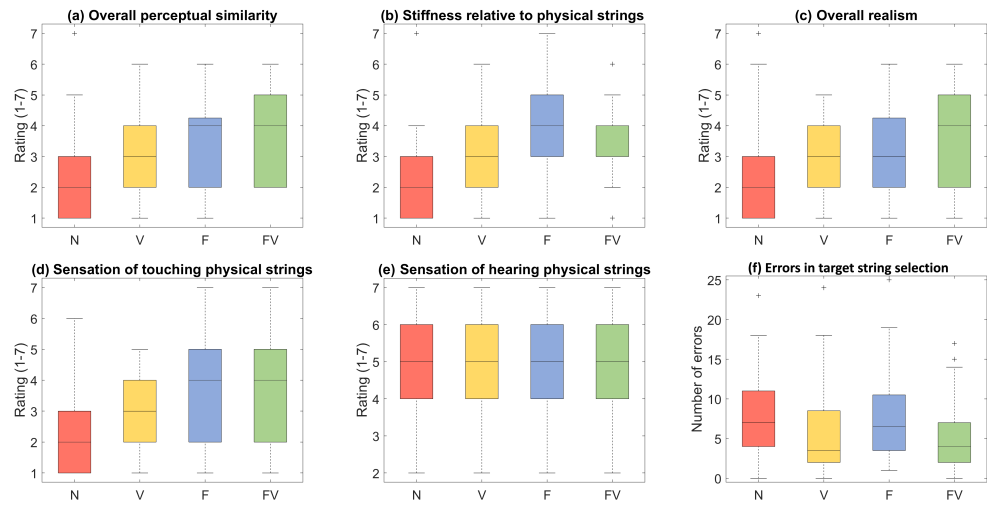


Figure 6. Keytar: experimental results.

4. Bowing

The best-known category in frictional musical instruments is that of bowed strings, with many instruments worldwide sharing this type of string excitation and being made of various materials. Typically these strings are attached to resonant bodies completing the instrument such as in the violin, viola, or cello, all essential elements in Western music ensembles, as well as other resonators such as in the square-shaped khuur of traditional Mongolian music or the diamond-shaped mesenqo of the Ethiopian musical tradition. In physical-based sound synthesis implementations, these bodies can be realized by a finite impulse response filter processing the sound output from a bowed string model. Onofrei et al. (2022b) proposed research aiming to offer realistic and natural synthesis of physical-based bowed string sounds through haptic feedback control of the strings' elasticity via frictional interaction. This project and its relevance to the field of musical haptics is described in the following subsections.

4.1. Synthesis Model

The mechanics of a damped stiff string of length L [m] with a circular cross section, defined for time $t \geq 0$ [s] and space $x \in \mathcal{D}$ [m] within the domain $\mathcal{D} = [0, L]$, subjected to a bowing excitation resulting in the frictional force F_{fr} [N], can be described by the following partial differential equation:

$$\partial_t^2 u = c^2 \partial_x^2 u - \kappa^2 \partial_x^4 u - 2\sigma_0 \partial_t u + 2\sigma_1 \partial_t \partial_x^2 u - \frac{F_{fr}}{\rho A}, \quad (1)$$

with the state variable $u = u(x, t)$ [m] representing the transverse displacement of the string. ∂_t and ∂_x describe a derivative with respect to time and space, respectively. The remaining parameters are as follows: wave speed $c = \sqrt{T/\rho A}$ [m/s] with tension T [N], material density ρ [kg/m³], cross-sectional area $A = \pi r^2$ [m²], radius r [m], stiffness coefficient $\kappa = \sqrt{EI/\rho A}$ [m²/s], Young's modulus E [Pa], area momentum of inertia $I = \pi r^4/4$, frequency-independent loss coefficient σ_0 [s⁻¹], and frequency-dependent loss coefficient σ_1 [m²/s].

What the equation essentially says is that the displacement of a string at any point, along its length and across time, is related to its level and type of external excitation, as well as to its material and physical properties.

The bow frictional force excitation F_{fr} [N], i.e., the external excitation, is modeled using the following friction model (Bilbao 2009):

$$F_{fr} = \delta(x - x_B) f_B \Phi(v_{rel}, a), \quad (2)$$

stating that an externally supplied bowing force f_B [N], localized at the bowing point x_B [m] along the length $x \in [0, L]$ by means of a Dirac function $\delta(x - x_B)$, is scaled using a nonlinear function $\Phi(v_{rel}, a)$ which gives the friction characteristic. This function depends on the relative velocity v_{rel} [m/s] between the bow and the string at the bowing location, and on a friction parameter a [s²/m²], which gives the shape of this nonlinearity and essentially tunes the frictional behavior.

The second externally supplied input to the model, in addition to the bowing force f_B , is the velocity of the bow v_B [m/s], which directly contributes to the relative velocity between the bow and the string:

$$v_{rel} = v_{rel}(t) = \partial_t u(x_B, t) - v_B. \quad (3)$$

Lastly, the string is assumed to be simply supported at its boundaries such that:

$$u = \partial_x^2 u = 0, \quad \text{at } x = 0, L. \quad (4)$$

This model can be numerically simulated using finite difference time domain methods, such that the transverse displacement of the string $u(x, t)$ can be approximated under any type of excitation governed by external inputs, i.e., the bowing force f_B , bowing velocity v_B , and bowing location along the string x_B . Details of the implementation of this numerical solution can be found in [Onofrei et al. \(2022b\)](#).

One reason for such a detailed model is that it affords a natural interaction, especially reactive to the inputs from the user.

4.2. Control Interface

The primary objectives of the control interface for the bowed string physical model were to offer flexibility with regards to the instrumental gesture motion and to enable realistic instantaneous haptic feedback in terms of both the elasticity of the strings as well as the frictional interaction between the bow and the string.

Another professional haptic device from the range of products available from 3D Systems was found suitable for these objectives: 3D Systems Touch, illustrated in [Figure 1](#), together with its local coordinate system (CSYS) and the pivot joints B1, B2, and B3 which enable the translation in x , y , z directions of the gimbal joint (equivalent to pivot joint B2) and the three possible rotations of the stylus pen. It is a slightly lower-end version compared to the Touch X device used to control the mass–spring–dampers model described in the previous section, but still offers highly accurate force feedback at the gimbal position, through very silent internal motors located at the joints A1, A2, and A3. The pen-shaped stylus and its flexibility of motion afforded by the device's system of joints allow for an accurate reproduction of the bowing gestural motion, with the user being able to hold the pen in a similar way to holding a real bow.

4.3. Real-Time Application

Building on the experience afforded by the rubbed mass–springs project described in [Onofrei et al. \(2022a\)](#), and also summarized in [Section 5](#) in this work, it was chosen to continue with the same development environment for bowed strings audio-haptic musical application, i.e., C++ and the JUCE framework. The resulting code is open-source and available at [Onofrei \(2022a\)](#) and a demo video can be found at [Onofrei \(2022b\)](#).

Four simply supported damped stiff strings, whose mechanical behavior is described by the model given in [Section 4.1](#), are placed in the virtual 3D space, and in the corresponding haptic space of the touch device, in a trapezoidal cross-section, similar to how strings are placed on the neck of a violin. The violin's strings are tuned to fundamental frequencies, which correspond to the musical notes G3, D4, A4, and E5, respectively. The tuning strategy and usual parameters for such strings are described in [Willemssen \(2021\)](#). A perfectly rigid bow can excite the strings via the frictional interaction model described in [Section 4.1](#). This

virtual space's CSYS is the same as that of the Touch haptic device. Both the bow and the strings are represented by infinitely thin lines.

4.4. Graphical User Interface (GUI)

Figure 7 shows a snapshot of the app during operation, together with its external control.

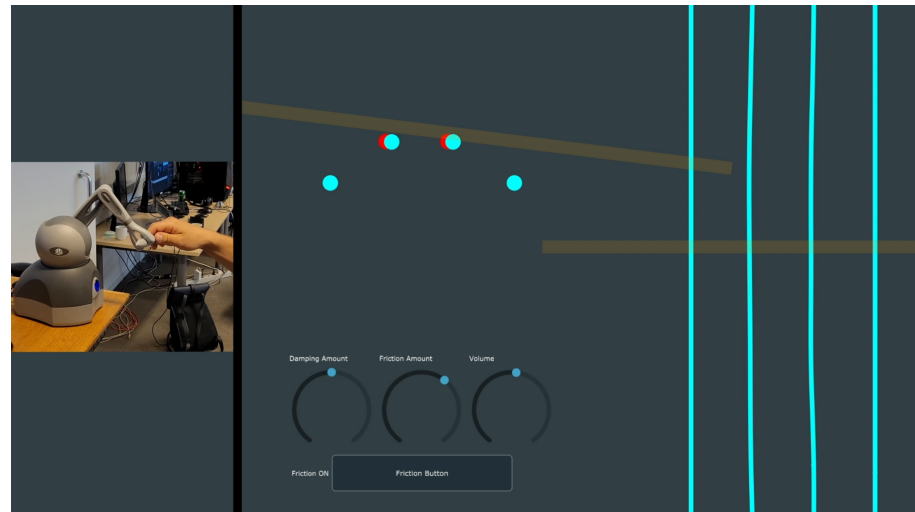


Figure 7. Snapshot from the demo video [Onofrei \(2022b\)](#) of the digital music instrument showing both the software application window as well as the control via the haptic device.

As it can be seen, the window is separated into three distinct sections. On the right, a top view of the strings, i.e., the strings projected onto the xz horizontal plane, can be seen. The long transparent orange rectangle represents the bow, whose color changes to gray when not in contact with any of the strings, such as when hovering over them in the virtual 3D space. When in contact, however, the opacity of its orange color is directly proportional to the externally supplied bow force, f_B . This means that the bow appears to be more transparent with a smaller bowing force, giving additional visual feedback to the control. Another part of the visual feedback is that the displacement of the strings is updated in the GUI at a rate of 24 frames per second, as a compromise of CPU use by the graphics thread and close to the frame rate of human visual perception. In the top-left corner area of the app window, the strings are visualized as projected in the frontal plane xy . Here, the trapezoidal layout of the strings can be seen. Additionally, the maximum displacement of the strings in this projection view is illustrated as red ellipses placed underneath the static cyan ones, which represent the “at rest” position. The last section of the app window is in the bottom left corner, where there are a number of knobs that control various parameters of the physical model, as well as a button that toggles the frictional haptic feedback in the control interaction on and off. The first knob adjusts the damping amount, which is a combination of the frequency in-dependent damping, $\sigma_0 \in [0.1, 10]$, and the frequency dependent damping, $\sigma_1 \in [0.0001, 0.015]$, as described in Section 4.1. The second knob controls the friction parameter, $a \in [0.01, 15,000]$, which is logarithmically mapped so that the friction interaction between the bow and the strings transitions smoothly from “slippery” to “sticky.” Finally, a global volume-gain knob is included to control the app’s master volume.

4.5. Haptic Mapping

As indicated previously, the stylus of the touch device is an ideal interface for controlling a virtual bow. The center of the bow is mapped to the position of the gimbal joint in the virtual physical model space. For ease of control, the orientation of the bow in the physical model is fixed to be parallel to the frontal plane, or xy , but permitted to rotate

relative to the horizontal plane, xz . In the horizontal plane projection, the bow is always perpendicular to the strings. Movement of the stylus in the haptic space corresponds to movement of the bow in the physical model space.

At each haptic frame, the shortest distance between the bow and each string is determined as the shortest line/segment perpendicular to both elements. It is also possible to determine the precise coordinates of this segment's end points, keeping track of which of the two end points is on the string and which one lies on the bow. Using this information, it is possible to know when the bow is pushing on either of the strings and also calculate a penetration distance relative to each string, i.e., the shortest distance from the bow to each string in its "at rest" position. A haptic feedback can then be sent to the gimbal joint point consisting of an elastic force determined from the multiplication of an heuristically chosen spring stiffness with the magnitude of the penetration distance. This is how the user can feel the elasticity of each of the strings in the virtual space. When the bow is in touch with multiple strings, the aggregate of their reaction forces is provided as haptic feedback. Through linear mapping, this force is directly proportional to the externally provided bow force, f_B , meaning that the force with which the user presses onto the haptic strings gives the first input of the physical model. The second input, i.e., the externally supplied bow velocity v_B [m/s], is given by the velocity of the stylus of the haptic device, linearly mapped to some reasonable physical range of bowing velocities, chosen to be $[-0.4, 0.4]$ m/s.

During the interaction of the bow with any of the strings, the frictional force resulting from the physical model is sent as haptic feedback to the gimbal joint, split in x , y , z components given by the opposite direction of motion of the bow. As for the case of the elastic feedback, frictional forces resulting from multiple synchronous interactions are summed up.

4.6. Informal Evaluation

A formal evaluation of the musical interface described above has not yet been carried out. However, qualitative feedback has been given by users of the device on two occasions. The first was given during a visit of a group of employees of the Danish Music Museum to the multimedia experience lab of Aalborg University, and the second at the International Congress of Acoustics (ICA2022), where between five and ten conference attendees used the device following the presentation of the companion paper at the event.

In the first instance, the museum employees were all positively impressed by the realism of the interaction as well as the resulting sound. This is due to the careful mapping of both the parameters of the physical model such as the material properties of the string and the range of values of the friction parameter, which governed the friction interaction and also the mapping of the externally supplied inputs to the model, i.e., the bow force f_B and the bow velocity v_B . Heuristic calibration of such physical models is a necessity, akin to the fine tuning finishing touches an instrument maker carries out.

The feedback from the acoustics conference attendees was more technical, with users reporting that, with the haptic feedback turned on, varying the attack and the timbre of the sound became more natural and could be more easily steered in the desired direction. In addition, the real-time visual feedback shown in the GUI helps users better evaluate their movements in the 3D virtual space.

5. Rubbing

Mass-spring systems are the typical example for mechanical harmonic oscillators, simple resonators that vibrate infinitely at a single natural frequency, unless a damper is added to the system which controls the decay of the sound. They can however still produce sounds with rich timbre quality, depending on their excitation mechanism. A virtual musical instrument based on a physical model of a bank of such mass-spring-dampers excited via rubbing has been developed and presented in [Onofrei et al. \(2022a\)](#). Three control strategies/devices have been proposed, which allow for an increasingly realistic and multimodal interaction between the performer and instrument. A description of

the audio-haptic physical synthesis model is given in the next subsection, followed by a presentation of the aforementioned controls with details regarding the chosen mapping strategies. Finally a presentation of a user study aiming to compare the experience of performing with the different interfaces is offered in the last subsection.

5.1. Synthesis Model

As shown in Figure 8, the model consists of a mass m [kg] coupled to a rigid support via a parallel system comprising a linear spring with spring constant k [N/m] and linear damper with damping coefficient c [kg/s]. This mass is then excited by a rigid and weightless stick that is placed onto it with normal force N [N] and moved at a rubbing velocity v_r [m/s]. Its displacement relative to its resting position at time t [s] is denoted as $u(t)$ [m] and is described by the following ordinary differential equation (ODE) of the second order:

$$m\partial_t^2 u + c\partial_t u + ku = -F_{fr}, \quad (5)$$

where, F_{fr} [N] is the nonlinear frictional force at the point of contact between the stick and the mass, modelled as:

$$F_{fr} = N\phi(v_{rel}, a). \quad (6)$$

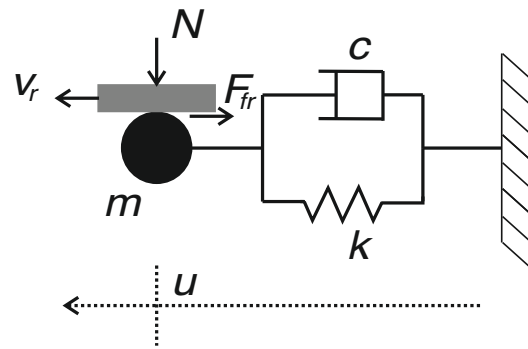


Figure 8. Mass-spring-damper system excited via friction.

This equation scales the applied normal force, N , by a nonlinear function, $\phi(v_{rel}, a)$, which depends on the relative velocity between the stick and the mass, v_{rel} , and a friction parameter, a , that determines the shape of the function. This shape is illustrated for various values of a in Figure 9, from which it can be seen that the model interpolates between a viscous friction model (for small values of a) and a Stribeck model (for large values of a). Modifying this parameter therefore changes the “stickiness” of the interaction, and can critically alter the perception of friction. This model was first introduced by Bilbao (2009) and it has the benefit of being continuous and differentiable, allowing for the non-linearity to be more easily to computed numerically. It has been used for real-time audio simulations of bowed strings due to its stability, small number of parameters, and ease of use.

The un-damped system’s natural resonance frequency, f_0 , is given by:

$$f_0 = \frac{1}{2\pi} \sqrt{k/m}. \quad (7)$$

Damping will somewhat skew this figure; however, it only affects resonance frequency near the critical damping, which is not relevant for musical applications.

Using a numerical simulation approach known as finite difference time-domain analysis, a solution to this mathematical model, i.e., determining the motion of the mass under a particular external excitation, can be obtained. The specifics of this can be found in Onofrei et al. (2022a), but in summary, the system is discretized in the temporal and spatial domains, and a system of equations can be set up to calculate the displacement of the mass at a future time index given the displacement and external input values of the system at previous time indices. The audio output is then determined by the computed mass velocity,

which is proportional to the sound generated by a physical instrument and preserves high-frequency information.

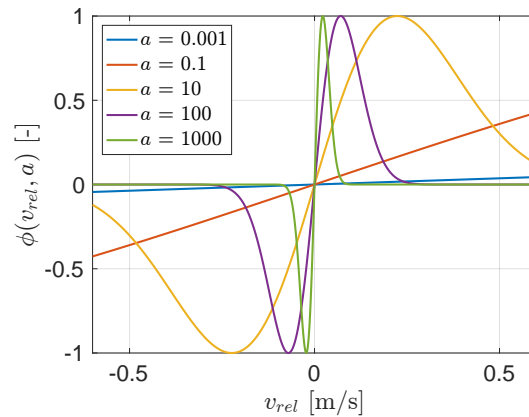


Figure 9. Shape of nonlinear friction characteristic $\phi(v_{rel}, a)$ for various values of a [s^2/m^2].

5.2. Real-Time Application

The objective was to develop a real-time polyphonic audio application comprised of a bank of mass-spring-dampers tuned to different musical frequencies. In addition, it was needed to have dynamic control over the damping and friction parameter a , which regulates the interaction's “stickiness”.

The real-time application was developed using the JUCE framework and written in C++. A demo video is available at [Onofrei \(2022c\)](#), and Figure 10 depicts a screenshot of the application in operation, which runs at a sampling frequency $f_s = 44,100$ Hz. Eleven mass elements are positioned in the center of the application window and appear as red-filled circles. The lower the natural frequency, the darker the tone. They are tuned with a continuous interval of 27.5 Hz from a resonant frequency of 55 Hz up to 330 Hz, but other musical scales can easily be chosen in a recompiled version. The stick which is used to excite the masses is depicted as a long, variable-opacity rectangle. When it is not in contact with the masses, it is depicted as gray, and as orange when it is. Increased opacity is proportional to the amount of normal force applied to the masses by the stick, N . The displacement of the masses resulting from the simulated physical interaction is updated at a rate of 24 frames per second in the application window, giving the user with visible feedback. The rubbing velocity, v_r , is given by the longitudinal velocity of the stick.

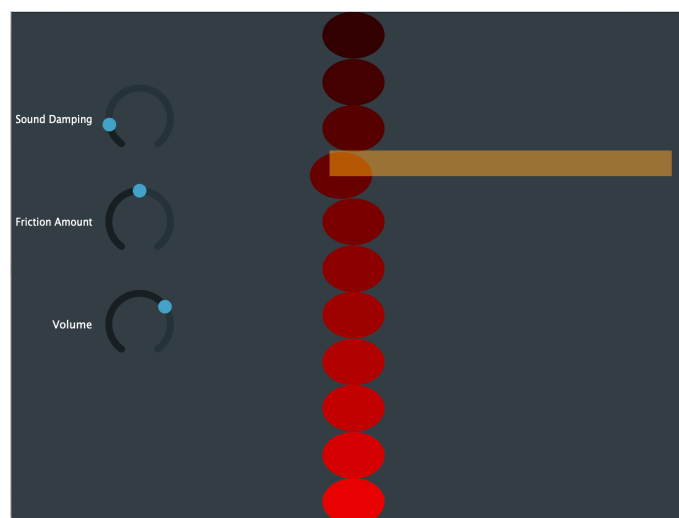


Figure 10. Snapshot of the audio application window. The stick (yellow rectangle) is exciting one of the masses (red circles), which is displaced from its rest position.

5.3. Control Strategies

Three control strategies, each using a difference interaction device, were developed for the application with the intention of mimicking with increasing precision the gesture of rubbing a mass with a stick.

First, the app was controlled using the mouse, the universal virtual musical instrument controller. The mouse left click was used to bring the stick in contact with the masses and the normal force N could be adjusted in real-time using the mouse wheel. Then, the longitudinal velocity of the mouse drag was mapped to the rubbing velocity v_r . This gesture is reminiscent of the desired “rubbing” musical gesture.

The second control method used Sensel Morph, a tablet-sized pressure-sensitive touchpad that is sensitive and responsive. This control integrated the pressing action into the instrumental gesture, which can be mapped to the rubbing normal force. The user can now control this excitation parameter more dynamically. Unlike the mouse control, the Sensel’s binary mapping for touching the masses cannot be manually toggled. It was chosen to map the stick’s position to where a user’s finger first contacts the gadget. A second finger signifies that the stick must touch a mass, with the normal force depending on the applied pressure.

The final control strategy aimed to both simulate the instrumental gesture and provide realistic haptic feedback based on the physical model, thus enhancing the multisensory interaction with the audio application. This control could be achieved by using the 3D Systems Touch X haptic device, which is illustrated in Figure 1 together with a reference coordinate system, used in the following descriptions. It is a pen-shaped robotic arm with six degrees of freedom of motion and a set of motors that can offer three degrees of freedom of force feedback.

The location of the stick in the application window was mapped to the position of the pen tip in the x - z plane. By providing two rigid haptic x - y planes along the z axis, the arm’s mobility in the z direction was constrained, simulating the sensation of striking a wall. Another haptic plane parallel to x - z with elevation $y = 0$ was modeled. However, this plane is elastic and provides force feedback proportional to y -direction penetration, which corresponds to negative y coordinate values. When y is greater than zero, which corresponds to when the stick is hovering above the masses, there is no feedback. At $y = 0$, the pen tip makes contact with the masses. The rubbing normal force, N , is then mapped to the force feedback experienced in the y direction, originating from the elastic plane. Lastly, the x -direction movement velocity of the pen tip is mapped to the rubbing velocity v_r . Furthermore, in order to strengthen the realism of “rubbing”, the physical model simulation’s computed friction force is mapped to the force feedback in the x direction. Users can, in this manner, truly feel the stick-slip interaction between the stick and the masses.

5.4. Evaluation

A user study was conducted to evaluate the overall experience of using the virtual instrument with the different control strategies introduced in the previous subsection. The methodology and results of this evaluation are presented in more detail in [Onofrei et al. \(2022a\)](#) and an overview and conclusions are provided in this current section. The main goal was to determine whether imitating the instrumental gesture enhances this experience. Towards this objective, both quantitative and qualitative methods were employed. Specifically, a questionnaire composed of seven statements, each related to a single attribute/criteria, rated on a seven-point Likert scale, from “Strongly Disagree” to “Strongly Agree”, was filled out by the subjects at the end of their session with the app using its different controls. Then, an interview was conducted to see whether participants understood what the application’s various controls did and how they affected the sound.

The evaluation session consisted of the subjects, following a brief introduction of each control, freely exploring the app and its sound with the aforementioned controls, all

available at the same time. It usually lasted between 10 and 30 min, depending on the choice of each participant.

The questionnaire items and their associated attributes are given in Table 1 and an overview of the results can be seen in Figure 11, which shows a boxplot visualization of the users’ replies to the items, achieved by giving an ordinal value to each point in the Likert scale.

Table 1. The questionnaire items with corresponding anchors of the seven-point Likert scale (Strongly Disagree–Strongly Agree).

Attribute	Statement
1. <i>intuitiveness</i>	The interaction with the objects was intuitive.
2. <i>playability</i>	I obtained sounds as by my intentions.
3. <i>enjoyment</i>	The interaction was fun.
4. <i>expressiveness</i>	I obtained enough types of sounds.
5. <i>difficulty</i>	It was easy to obtain the same sound twice.
6. <i>realism</i>	The interaction was realistic.
7. <i>precision</i>	The interaction was precise.

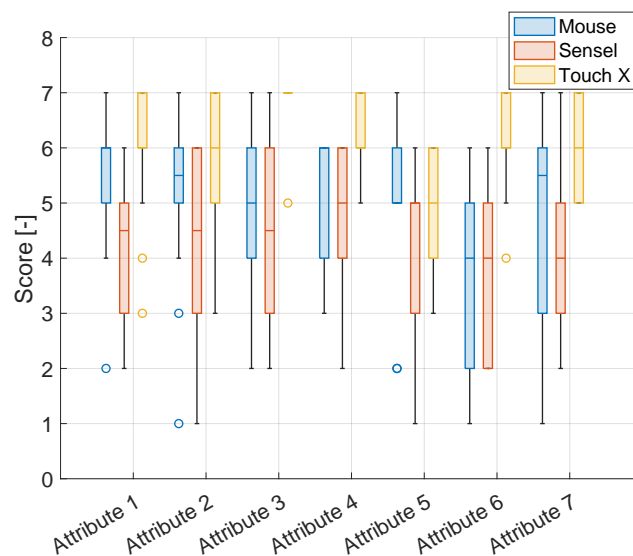


Figure 11. Boxplot visualizing the results related to the 7 questionnaire statements for the three control devices.

The Touch X routinely outperforms the other two controls. When it comes to *enjoyment*, the users almost unanimously “Strongly Agree” that the interaction was fun, with only a single score of “Agree”. The items *expressiveness*, *realism*, and *precision* also received only positive evaluations. Where the Touch X scored lowest was on the difficulty statement, which was expected, since none of the participants had used such a device before and a learning curve was to be expected. One somewhat unexpected result was that the mouse control was generally preferred over the Sensel. One possible hypothesis for why this occurred, in addition to the inherent familiarity of users with the mouse, is that it also provides passive haptic feedback in the form of button clicks or wheel use, which may better inform users when their controls are applied in the interaction.

As a result of the qualitative assessment, i.e., the interview with the participants regarding their experience, further significant insights were discovered. It was discovered that frictional haptic feedback helped users understand the effect of the “Friction Amount” knob, which influenced the “stickiness” of the contact. One user remarked that the knob “alters perceptions of hardness or viscosity ... like stirring soup”. Indeed, with low-friction

parameter values, one must use more force to excite the masses, although this is not the case with high values, which correspond to adhesive interactions. Here, rubbing does in fact feel more viscous (one of the definitions of the word being “having a thick or sticky consistency”).

6. Discussion

Our projects highlight the benefits of design for digital musical instruments that, while providing natural haptic feedback, preserve the primitive gestures dictated by the respective physical model. As mentioned in the introduction, physical modeling synthesis is a fruitful technique for this merger, as auditory and haptic feedback are inherently connected. In spite of sharing the same haptic device, the three digital music interfaces show evident differences in their model and consequent implementation.

The keytar represents a successful case of digital music interface realization on a programming environment specifically intended for, and hence accessible to, interaction designers. For this reason, the development of the keytar focused primarily on prototyping its components based on simple physical interpretations and, on the implementation side, with relatively less attention to code the efficiency and robustness of the overall software architecture, both autonomously managed by Unity3D. Some issues caused by this design approach, especially affecting the haptic feedback, were solved using creative workarounds. Further shortcomings due to inaccurate contact detection, resulting in occasional system instability, promise (if not already at the time when this paper is written) to be solved by the new physics engine and Unity3D asset releases. In the end, the keytar provides interaction designers with a perceptually validated environment where one can visually edit most features of a plucked string instrument including size, positioning, plucking points, intonation, and timbre, as well as the physical properties of the string set. Thanks to its haptic components, it exposes a rich palette of physical material properties for manipulating not only the sound, but also the tactile feeling of the instrument strings, making the keytar an ideal candidate for the virtual testing of novel plucked-based interactions. While experimenting new plucked musical interactions, designers with a focus on the arts can rely on the testbed summarized by Figure 6. This testbed, whose results align well with existing findings in musical haptics [Papetti and Saitis \(2018\)](#), provides one with the dependencies of the interface users’ objective feelings, qualitative impressions, and performance on the type and amount of feedback. Along with its mentioned technical features, in the end the keytar forms a platform that can support the artistic design of plucked interactions.

Bowing required a more careful design of the physical model at core level. Realization in an environment as JUCE inherently requires a longer development time, as part of a design plan that must be started with a robust formalization of the dynamical system. Such a plan prevents one from testing modeling ideas on the fly, as Unity3D often allows. The result of this activity is a procedure written in native code that, if correctly implementing the formulas specifying the system, gives rise to an efficient application module applicable to different digital music interfaces. The flexibility of the application works in principle yet, in practice, it can hardly be realized without customizing the module to the new interface, as a software architecture generated by Unity3D allows for. Nevertheless, the rubbing model shown in Section 5 was extended to bowing with certainly far less effort than coding a new model would have required.

Taken together, ignoring such design aspects and implementation issues, plucking, bowing, and rubbing give a significant picture about the importance of providing performers with gestural controls going far beyond the expressive possibilities of conventional input/output computer devices. As mentioned in the introduction, this issue became immediately clear to the computer music community researching new digital instruments. Part of this community in particular realized that a successful extension of the control space would not be possible without *remutualizing* the instrument, as result of an evolutionary effort jointly made by performers (i.e., artists) and craftsmen (i.e., interaction designers) [Cook \(2004\)](#). The potential benefits of an increased attention to digital instrument control

as an integrative, indistinguishable part of the player–controller–synthesizer system are enormous and still partially unexplored, as testified by, for example, the excellent quality of string sounds obtained by McPherson et al. by driving a sound synthesis algorithm as simple as Karplus–Strong with acceleration data at a high rate acquired using metal strings [Jack et al. \(2020\)](#).

Some specific factors about the role of the visual interface and its implications to multisensory sonic interaction design are worth consideration. In the case of the keytar, Unity3D in fact allowed for the fast development of a relatively accurate 3D scene providing, on top of a pleasant background populated with musical objects, abundant contextual information to the performer about the physical interaction. Conversely, JUCE allows for the inclusion of standard graphic libraries that make it harder to develop a rich visual scenario, rather suggesting the developer focuses on the creation of primitive, preferably 2D objects. However, if rationally designed, such objects are not less informative about the interaction context and, in some cases, they can describe the essence of a dynamic situation even more precisely although with less pictorial appeal.

We remain noncommittal about which graphic design approach fits better the development of a digital music interface. Rather than trading between the richness and essence of a scene, our suggestion to the reader is to target embodiment as the ultimate, artistic goal of a multisensory design approach to digital instruments. As mentioned in the introduction, a key aspect of technology-mediated embodied music cognition [Schivavo and Menin \(2013\)](#) in fact concerns the transfer of energy between the digital instrument and the performer. If in the past the only physical carrier enabling this transfer was sound (indeed, this carrier is common to digital and traditional instruments) haptic devices for the first time allow performers to interchange energy with digital music interfaces through *real* force patterns. Consequently, these devices prevent us from delegating such patterns to indirect representations carried through vision, and the sensory haptic illusions that a careful design of the graphic interface can elicit also in a virtual performance scenario [Hachisu et al. \(2011\)](#); [Parviainen \(2012\)](#); [Peters \(2013\)](#).

In front of the goal to embody the interaction afforded by a digital music instrument, the Unity3D developer may opt for exporting the 3D scene to a head-mounted display using the powerful assets available in that programming environment for this purpose. As seen in Section 3.4, this idea was experimented using the keytar. More visionary, and also at reach of JUCE-based realizations, is the idea of substituting the graphic interface with a visual proxy located on the interaction point. While participants to the informal evaluation of the bowing project presented in Section 4.1 declared they were helped by visual feedback, no obstacles in principle exist to re-designing this feedback in ways to augment the presence of the interaction touchpoint without the help of a screen or even a head-mounted display.

The development of this and other ideas are left to discussions that need scientists and artists to intersect together in an effort to find a common ground. Ultimately, this paper has been written for inspiring such an effort.

7. Conclusions

Together, the three projects have illustrated a variety of opportunities around the multisensory interaction with physically based musical resonators. In spite of the constraints existing especially on hardware devices and computing resources, it has been shown that different design approaches lead to different interactions modeling both real and unconventional musical gestures. In this sense, they proved what was initially hypothesized, i.e., that the inclusion in the interface of programmable force feedback devices can play a significant role in increasing the quality of the interactions with a physics-based digital string, with possibilities for the extension of such interactions to unexplored musical instrument controls.

Realizing this required different software development strategies and skills, all however united by creative thinking as a necessary premise. While writing this paper, the

authors have tried to transmit the importance of creativity as an attitude that must ever be present in the interaction designer before any technical skill, which is needed at a later stage as a requirement to design usable digital music interfaces.

Author Contributions: Conceptualization, M.G.O. and F.F.; methodology, M.G.O. and F.F.; software, M.G.O. and F.F.; validation, M.G.O., F.F. and S.S.; formal analysis, M.G.O.; investigation, M.G.O., F.F. and S.S.; resources, M.G.O., F.F. and S.S.; data curation, M.G.O., F.F. and S.S.; writing—original draft preparation, M.G.O. and F.F.; writing—review and editing, M.G.O., F.F. and S.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to null ethical implications of the experiments in which participants were involved.

Informed Consent Statement: Patient consent was waived for this study, due to null effects of the experiments in which participants were involved.

Data Availability Statement: Data supporting reported results can be found at the Authors' institutions upon request to the Authors.

Acknowledgments: The Authors acknowledge work from Niels Christian Nilsson, Razvan Paisa, Andrea Passalenti and Roberto Ranon.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Behringer. 2021. BCR2000 MIDI Controller. Available online: <https://www.behringer.com/product.html?modelCode=P0245> (accessed on 25 March 2021).
- Berdahl, Edgar, Bill Verplank, Julius O. Smith, III, and Günter Niemeyer. 2007. A physically intuitive haptic drumstick. Paper presented at the International Computer Music Conference (ICMC), Copenhagen, Denmark, August 27–31; pp. 150–55.
- Bilbao, Stefan. 2009. *Numerical Sound Synthesis*. Hoboken: John Wiley and Sons, Ltd.
- Cook, Perry R. 2004. Remutualizing the musical instrument: Co-design of synthesis algorithms and controllers. *Journal of New Music Research* 33: 315–20. [\[CrossRef\]](#)
- Figueiredo, Lucas Silva, João Marcelo Xavier Natario Teixeira, Aline Silveira Cavalcanti, Veronica Teichrieb, and Judith Kelner. 2009. An open-source framework for air guitar games. Paper presented at the VIII Brazilian Symposium on Games and Digital Entertainment, Rio de Janeiro, Brazil, October 8–10; pp. 74–82.
- Fontana, Federico, Andrea Passalenti, Stefania Serafin, and Razvan Paisa. 2019. Keytar: Melodic control of multisensory feedback from virtual strings. Paper presented at the Conference on Digital Audio Effects (DAFx-19), Birmingham, UK, September 2–6.
- Fontana, Federico, Razvan Paisa, Roberto Ranon, and Stefania Serafin. 2020. Multisensory plucked instrument modeling in unity3d: From keytar to accurate string prototyping. *Applied Sciences* 10: 1452. [\[CrossRef\]](#)
- Fontana, Federico, Stefano Papetti, Hanna Järveläinen, and Federico Avanzini. 2017. Detection of keyboard vibrations and effects on perceived piano quality. *Journal of the Acoustical Society of America* 142: 2953–67. [\[CrossRef\]](#) [\[PubMed\]](#)
- Georgiou, Orestis, William Frier, Euan Freeman, Claudio Pacchierotti, and Takayuki Hoshi, eds. 2022. *Ultrasound Mid-Air Haptics for Touchless Interfaces*. Human–Computer Interaction Series; Cham: Springer Nature. [\[CrossRef\]](#)
- Gorman, Mikhail, Amir Lahav, Elliot Saltzman, and Margrit Betke. 2007. A camera-based music-making tool for physical rehabilitation. *Computer Music Journal* 31: 39–53. [\[CrossRef\]](#)
- Guettler, Knut, Erwin Schoonderwaldt, and Anders Askenfelt. 2003. Bow speed or bowing position—which one influences spectrum the most? Paper presented at the Stockholm Music Acoustics Conference (SMAC 03), Stockholm, Sweden, August 6–9.
- Guizzo, Erico. 2010. Keyboard maestro. *IEEE Spectrum* 47: 32–33. [\[CrossRef\]](#)
- Hachisu, Taku, Michi Sato, Shogo Fukushima, and Hiroyuki Kajimoto. 2011. Hachistick: Simulating haptic sensation on tablet pc for musical instruments application. Paper presented at the 24th Annual ACM symposium adjunct on User interface software and technology, Santa Barbara, CA, USA, October 16–19; pp. 73–74. [\[CrossRef\]](#)
- Hsu, Mu-Hsen, Wgcw Kumara, Timothy K. Shih, and Zixue Cheng. 2013. Spider king: Virtual musical instruments based on microsoft kinect. Paper presented at the International Joint Conference on Awareness Science and Technology Ubi-Media Computing (iCAST 2013 UMEDIA 2013), Aizuwakamatsu, Japan, November 2–4; pp. 707–13.
- Hwang, Inwook, Hyungki Son, and Jin Ryong Kim. 2017. Airpiano: Enhancing music playing experience in virtual reality with mid-air haptic feedback. Paper presented at the 2017 IEEE World Haptics Conference (WHC), Munich, Germany, June 6–9; pp. 213–18.
- Jack, Robert, Jacob Harrison, and Andrew McPherson. 2020. Digital musical instruments as research products. Paper presented at the International Conference on New Interfaces for Musical Expression (NIME2020), Birmingham, UK, July 21–25; Edited by Romain Michon and Franziska Schroeder. pp. 446–51. [\[CrossRef\]](#)

- Karjalainen, Matti, Teemu Mäki-Patola, Aki Kanerva, and Antti Huovilainen. 2006. Virtual air guitar. *Journal of the Audio Engineering Society* 54: 964–80.
- Karplus, Kevin, and Alex Strong. 1983. Digital Synthesis of Plucked String and Drum Timbres. *Computer Music Journal* 7: 43–55. [CrossRef]
- Leonard, James, and Claude Cadoz. 2015. Physical Modelling Concepts for a Collection of Multisensory Virtual Musical Instruments. Paper presented at the New Interfaces for Musical Expression (NIME), Baton Rouge, LA, USA, May 31–June 3; pp. 150–55.
- Liarokapis, Fotis. 2005. Augmented reality scenarios for guitar learning. Paper presented at the Theory and Practice of Computer Graphics (TCPG)—Eurographics UK Chapter, Canterbury, UK, June 15–17; pp. 163–70.
- Luciani, Annie, Jean-Loup Florens, Damien Couroussé, and Julien Castet. 2009. Ergotic Sounds: A New Way to Improve Playability, Believability and Presence of Virtual Musical Instruments. *Journal of New Music Research* 38: 309–23. [CrossRef]
- Manzo, Vj, and Dan Manzo. 2014. Game programming environments for musical interactions. In *College Music Symposium*. Missoula: College Music Society, vol. 54.
- Merchel, Sebastian, Ercan Altinsoy, and Maik Stamm. 2010. Tactile music instrument recognition for audio mixers. Paper presented at the 128th AES Convention, London, UK, May 22–25; New York: AES.
- Mihelj, Matjaž, and Janez Podobnik. 2012. *Haptics for Virtual Reality and Teleoperation*. Dordrecht: Springer Science. [CrossRef]
- Miller, Kiri. 2009. Schizophonic performance: Guitar hero, rock band, and virtual virtuosity. *Journal of the Society for American Music* 3: 395–429. [CrossRef]
- Miranda, Eduardo Reck, and Marcelo M. Wanderley. 2006. *New Digital Musical Instruments: Control and Interaction beyond the Keyboard*. Middleton: A-R Editions.
- Native Instruments. 2021. Traktor Kontrol S4. Available online: <https://www.native-instruments.com/en/products/traktor/dj-controllers/traktor-kontrol-s4/> (accessed on 25 March 2021).
- Nijs, Luc, Micheline Lesaffre, and M. Leman. 2009. The musical instrument as a natural extension of the musician. Paper presented at the 5th Conference of Interdisciplinary Musicology, Paris, France, September 26–29; pp. 132–33.
- Onofrei, Marius George. 2022a. Github Project. Available online: https://github.com/mariusono/Bowed_String_TouchCtrl (accessed on 26 December 2022).
- Onofrei, Marius George. 2022b. Youtube Demo Video. Available online: <https://youtu.be/uPHZMpQ5Z0k> (accessed on 26 December 2022).
- Onofrei, Marius George. 2022c. Youtube Demo Video. Available online: <https://www.youtube.com/watch?v=raLtqhdOYOM> (accessed on 26 December 2022).
- Onofrei, Marius George, Federico Fontana, and Stefania Serafin. 2022a. Rubbing a Physics Based Synthesis Model: From Mouse Control to Frictional Haptic Feedback. Paper presented at the 19th Sound and Music Computing Conference, St. Etienne, France, June 5–12. [CrossRef]
- Onofrei, Marius George, Federico Fontana, Stefania Serafin, and Willemsen Silvin. 2022b. Bowing virtual strings with realistic haptic feedback. Paper presented at the 24th International Congress of Acoustics, Gyeongju, South Korea, October 24–28.
- Pakarinen, Jyri, Tapio Puputti, and Vesa Välimäki. 2008. Virtual slide guitar. *Computer Music Journal* 32: 42–54. [CrossRef]
- Papetti, Stefano, and Charalampos Saitis, eds. 2018. *Musical Haptics*. Berlin: Springer International Publishing.
- Parviainen, Jaana. 2012. Seeing sound, hearing movement: Multimodal expression and haptic illusions in the virtual sonic environment. In *Bodily Expression in Electronic Music: Perspectives on Reclaiming Performativity*. Edited by Deniz Peters, Gerhard Eckel and Andreas Dorschel. London: Routledge. [CrossRef]
- Passalenti, Andrea, and Federico Fontana. 2018. Haptic interaction with guitar and bass virtual strings. Paper presented at the 15th Sound and Music Computing Conference (SMC 2018), Limassol, Cyprus, July 4–7; Edited by Anastasia Georgaki and Areti Andreopoulou. pp. 427–32.
- Passalenti, Andrea, Razvan Paisa, Niels C. Nilsson, Nikolaj S. Andersson, Federico Fontana, Rolf Nordahl, and Stefania Serafin. 2019. No strings attached: Force and vibrotactile feedback in a virtual guitar simulation. Paper presented at the IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, March 23–27; pp. 1116–17. [CrossRef]
- Peters, Deniz. 2013. Haptic illusions and imagined agency: Felt resistances in sonic experience. *Contemporary Music Review* 32: 51–164. [CrossRef]
- Pietrzak, Thomas, and Marcelo M. Wanderley. 2020. Haptic and audio interaction design. *Journal on Multimodal User Interfaces* 14: 231–33. [CrossRef]
- Rocchesso, Davide, Laura Ottaviani, Federico Fontana, and Federico Avanzini. 2003. Size, shape, and material properties of sound models. In *The Sounding Object*. Edited by Davide Rocchesso and Federico Fontana. Florence: Edizioni Mondo Estremo, pp. 95–110.
- Romero-Ángeles, Beatriz, Daniel Hernández-Campos, Guillermo Urriolagoitia-Sosa, Christopher René Torres-San Miguel, Rafael Rodríguez-Martínez, Jacobo Martínez-Reyes, Rosa Alicia Hernández-Vázquez, and Guillermo Urriolagoitia-Calderón. 2019. Design and manufacture of a forearm prosthesis by plastic 3d impression for a patient with transradial amputation applied for strum of a guitar. In *Engineering Design Applications*. Edited by Andreas Öchsner and Holm Altenbach. Cham: Springer International Publishing, pp. 97–121. [CrossRef]
- Rossing, Thomas D., ed. 2010. *The Science of String Instruments*. New York: Springer. [CrossRef]

- Schiavio, Andrea, and Damiano Menin. 2013. Embodied music cognition and mediation technology: A critical review. *Psychology of Music* 41: 804–14. [[CrossRef](#)]
- Schlessinger, Daniel, and Julius O. Smith. 2009. The kalichord: A physically modeled electro-acoustic plucked string instrument. Paper presented at the International Conference on New Interfaces for Musical Expression (NIME), Pittsburgh, PA, USA, June 4–6; pp. 98–101.
- Taylor, Tanasha, Shana Smith, and David Suh. 2007. A virtual harp with physical string vibrations in an augmented reality environment. Paper presented at the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (ASME), Las Vegas, NV, USA, September 4–7; pp. 1123–30.
- Willemsen, Silvin. 2021. The Emulated Ensemble: Real-Time Simulation of Musical Instruments Using Finite-Difference Time-Domain Methods. Ph.D. thesis, Aalborg University, Aalborg, Denmark.
- Wollman, Indiana, Claudia Fritz, and Jacques Poitevineau. 2014. Influence of vibrotactile feedback on some perceptual features of violins. *The Journal of the Acoustical Society of America* 136: 910–21. [[CrossRef](#)] [[PubMed](#)]
- Zhaparov, Meiram K., and Ulan Assanov. 2014. Augmented reality based on kazakh instrument “dombyra”. Paper presented at the 8th International Conference on Application of Information and Communication Technologies (AICT), Kazakhstan, Astana, October 15–17; pp. 1–4.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.