Developing block-movement, physical-model based objects for the Reactable

Smilen Dimitrov
Aalborg University
Copenhagen
Medialogy
sd@imi.aau.dk

Marcos Alonso
Universitat Pompeu Fabra
Music Technology Group
malonso@iua.upf.edu

Stefania Serafin
Aalborg University
Copenhagen
Medialogy
sts@imi.aau.dk

ABSTRACT

This paper reports on a Short-Term Scientific Mission (STSM) sponsored by SID European COST Action IC601.

Prototypes of objects for the novel instrument Reactable were developed, with the goal of study of sonification of movements on this platform. A physical model of a violin was used as an audio generation engine, which allowed development in two directions - a set of objects that expect motions similar to violin bowing, and a single object aiming to sonify contact friction sound. Informal evaluation was obtained from a Reactable expert user regarding these sets of objects. Experiments with these objects were also performed - related to both audio filtering, and interfacing with other objects for the Reactable.

Keywords

Reactable, physical model, motion sonification, contact friction

1. INTRODUCTION

The Reactable [6, 7, 8], developed by the Music Techology Group at University Pompeu Fabra (UPF) in Barcelona, is a novel electronic instrument, whose user interface is projected on a tabletop surface, and the users interact by moving and rotating objects placed on the table. In that sense the Reactable, features a rich and specific interaction language, both from a tactile and visual perspective. The Reactable is intended mostly for control of auditory devices typical in electronic music, such as sequencers, oscillators and LFOs. Each audio effect is represented by a physical object, marked with a unique fiducial pattern.

In this paper, we describe the research performed during a short term scientific mission (STSM) visit of the first author to UPF in Barcelona, which took place in January 2008. During such visit auditory feedback using physical models that aims to add an 'acoustic' behavior to the motions performed during interaction with a Reactable was investigated.

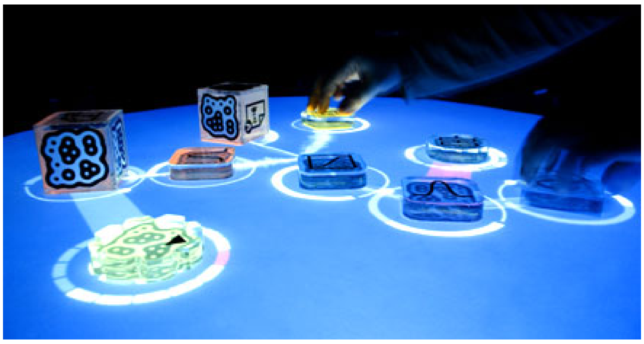

Figure 1: Interacting with a Reactable (from [10])

1.1 Reactable interaction language

Typically, a Reactable user interacts and creates sounds by moving and rotating different types of objects on the table; depending on the object types and their proximity to one another, links are established between them, which determine the audio flow. There are many types of Reactable objects - some represent sequencers, others represent samplers, envelopes, LFOs and other typical electronic music sound creation devices. This represents a part of the interaction language of the Reactable. In essence, each object can be manipulated through its rotation as one parameter, a 'finger' parameter (which is set by pressing the finger on the table in a proximity of an object), and through its position on the table. Although the links between objects are established by proximity, by briefly touching two objects it is possible to establish a 'hard link' between them, which doesn't break if another object is brought in proximity.

As these motions usually change some parameter of an electronic music instrument device, like filter amount or a sequence number, typically slow but precise motions are required from the player. This is also represented in other modes of electronic music performance, where sound is continuously generated, and the player only changes certain parameters of the individual instrument devices. In that sense, the Reactable does not exhibit 'acoustic' behaviour in a physical sense: due to its tactile nature, one can easily imagine a related table and objects made of a rough material; gliding of the objects upon the table surface, in this 'rough' case, would produce a contact friction sound, which lasts as long as the objects are in motion. To create an analogy with the real world, assuming that objects and table are made of, say, wood, it is easy to conceptualize that to produce significant amount of sound from this system, would require both a significant amount of force and motion from the player. From now on, we will name such motions 'block movements'.

2. METHOD

The work of developing objects that would demonstrate block motions on a Reactable, was made much easier by the efforts of the Reactable team, who provided a working and fully compatible standalone Reactable simulator for Windows, with an audio engine based in PureData (Pd) [11]. As it was relatively easy to build upon existing objects for inheritance of the user interaction, most of the work consisted of audio programming in Pd.

Since one of the defining high-level characteristics of block motion sound seems to be the relationship of sound volume to the velocity of the objects on the table, it was decided that the main parameter obtained from the objects, besides the standard parameters, would be the velocity of the objects, which could then be mapped to a sound parameter. Although, in principle, a contact friction sound is perceptually noisy, it could be generated through variuos sources [1].

A Pure Data real-time implementation of a physical model of frictional interaction between dry surfaces was available, which has already been described in [13, 14, 12]. It was decided that this friction model could be used as a sound generator for contact friction - especially in those ranges where high forces and low velocities would be involved. Due to the limited duration of the STSM visit, only a design and implementation of a prototype object was initally planned, to be followed by an expert user evaluation.

2.1 Mapping between the Reactable and the friction model

Figure 2 shows the first proposal for the mapping strategy between the Reactable and the friction model, called 'kviolin' in this simulation. The screenshot was extracted from the Reactable simulator. In the first proposal, four parameters of the friction model (velocity and position of the excitation, amplitude and frequency of the resonator) were selected and controlled by objects of the Reactable. The object called source controls the amplitude of the sound using a finger, and the excitation position using the rotation of the object, while the distance of the object from the center is mapped to fundamental frequency of the friction model. Excitation velocity and and force are kept constant. Upon adding the exciter object, it is possible to additionally control the exciter force and velocity. The exciter force is controlled through a finger parameter, whereas rotation of the object maps to a constant velocity.

This first proposal contradicted with common mapping strategies used in the Reactable, where frequency is related to rotation of an object. Moreover, the relatively slow camera used for tracking created some differences between the simulator and the tangible interface. Therefore a second mapping strategy was investigated, as shown in Figure 3. In this second strategy, the parameters mapped in the exciter object were switched, the constant velocity was mapped to a finger parameter and force was mapped to rotation. The second prototype was further improved, to produce the strategy

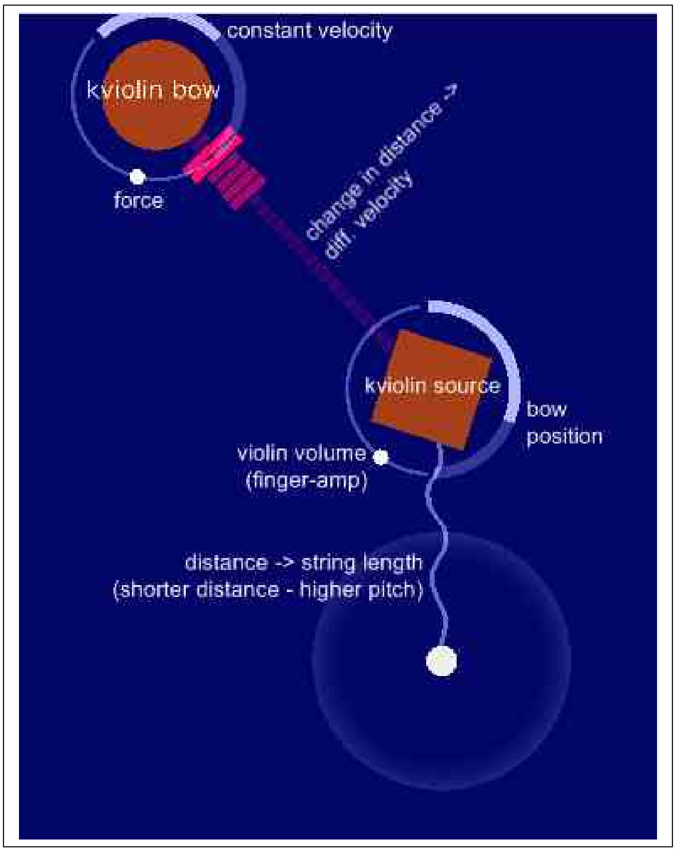

Figure 2: Reactable simulator showing the first proposal for a mapping strategy to connect the Reactable to the friction model.

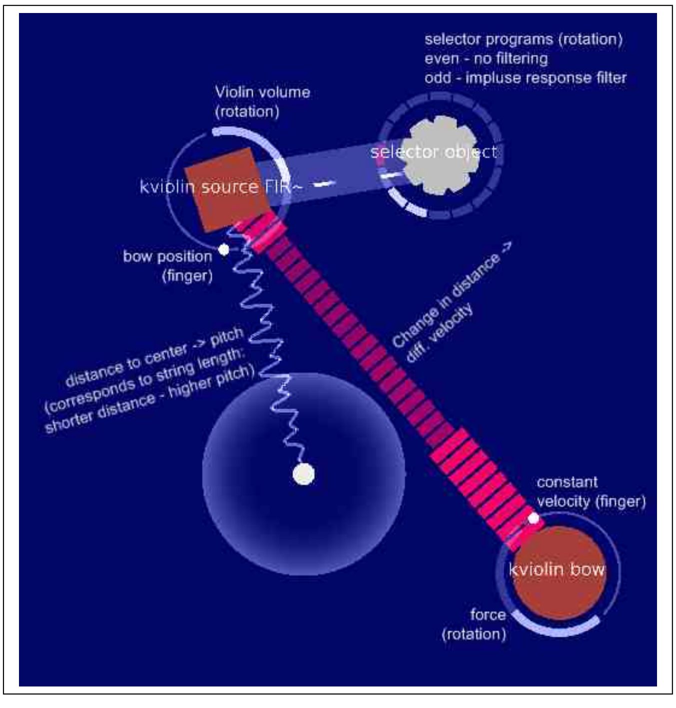

Figure 3: Reactable simulator showing the second proposal for a mapping strategy to connect the Reactable to the friction model.

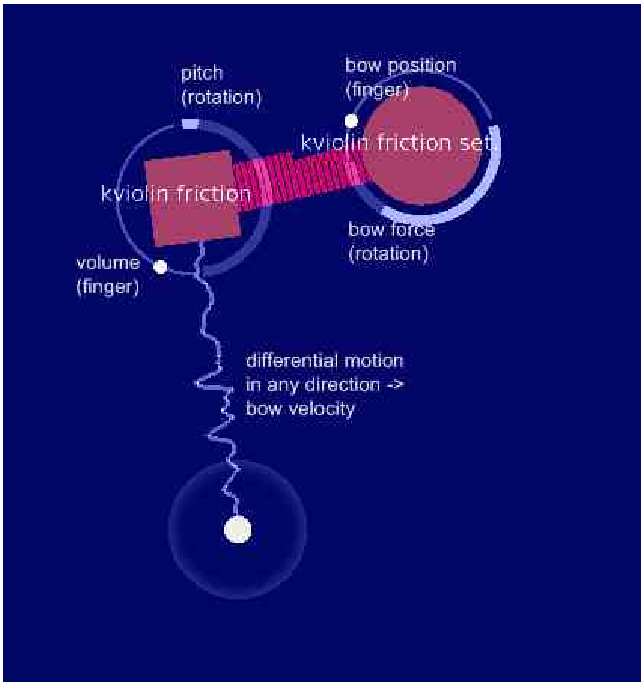

Figure 4: Reactable simulator showing the third proposal for a mapping strategy to connect the Reactable to the friction model.

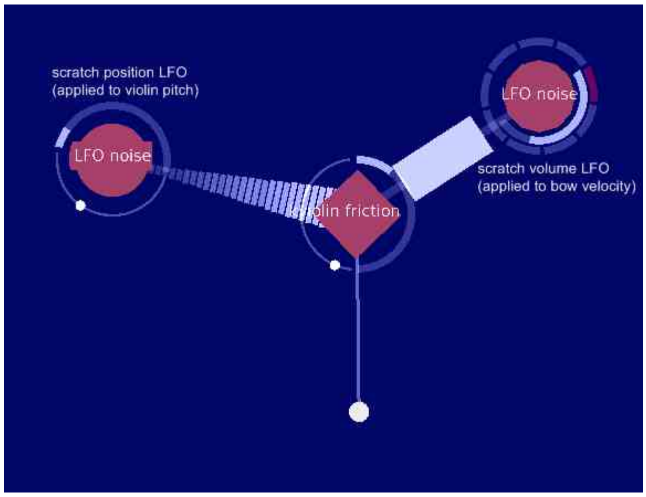

Figure 5: Reactable simulator showing the connection between friction models and scratch objects.

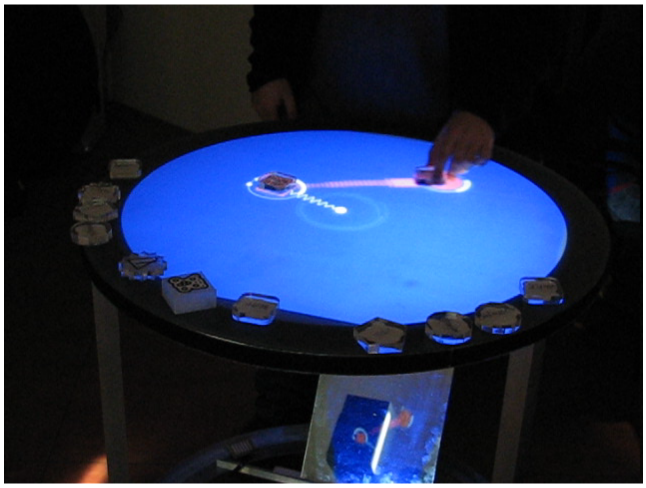

Figure 6: Friction objects for Reactable

Figure 7: Single surface friction object for Reactable

shown in Figure 4.

The implementation of the SkipProof (a DJ scratching application and virtual turntable developed by Hansen and others at KTH [5]) engine as a set of DJ scratching objects for the Reactable [2] was also furthered. As interfacing between the friction physical model and the SkipProof engine was attempted as a part of a previous STSM visit [3], it was attempted again - this time as an experiment in the context of Reactable objects. The connection between the two platforms is shown in Figure 5.

The original intent to develop a single block-motion object, changed soon after deciding to take upon the friction model as a sound engine base - as also in frictional interactions which happen in the real world, the motion of the interaction represents a block movement. Hence, the goal was extended with development of a prototype of a set of objects for the Reactable, where one would represent the interaction', and the other would represent the source. As as analogy to bowed string instruments, we can consider these objects as a 'bow' interacting with a 'string'.

3. RESULTS

The main results of the study visit are the production of prototypes of two sets of Reactable objects, and their preliminary (and informal) evaluation by an expert Reactable user.

The first set of objects is the pair of source and interaction objects aimed to emulate motions performed in real violins. The second set is a single object intended to simulate the sound of surface friction of moving objects in contact.

Additionally, video recordings were taken from some of the development tests; these, along with a development log, were posted online [4].

4. DISCUSSION

4.1 Reactable as a development platform

As mentioned previously, being the Reactable interface with Pd, the easiest way to create additional audio capabilities to it is by creating plugins for Pd. From a perspective of a new Reactable object developer, possibly the only glitch in the engine could be the current impossibility to set the so-called 'finger' parameters of Reactable objects directly from Pd (as it is possible with the 'rotation' parameter of the objects, for instance). Otherwise, it is relatively easy to develop the auditory behaviour of new objects using a Reactable simulator locally.

In our experiments we used a vision tracking system working at 60 fps. This created some problems with motion blur during fast motions. For future experiments, the motion blurring and low (in audio terms) 60 fps framerate must be taken into account - especially for objects that are to be moved in a faster, linear manner across the table. This proved to be a major difficulty in implementing a motion-based object, as for faster linear motions the system failed to detect the object, and the corresponding control signal used in audio was interrupted. Some measures were attemped to overcome this, which were not succesful, which finally resulted in the not-so-extatic evaluation of the Reactable object prototypes.

Here, that had to be taken into account for both the exciter object, and the single friction object, whose average velocity of motion across the table was used to derive a bow velocity signal. For these objects, accumulation of signal values, undersampling and linear smoothing was attempted to overcome the sudden change of values (during video tracking blurring). This, however, didn't prove to be efficient; averaging and low pass filtering in audio signal domain, would possibly be a much better approach to overcome these problems. On the other hand, one can try and avoid linear motions when designing interaction, and replace them with rotatory ones - as was suggested by the expert user. Although, it is important to note that the Reactable team currently works on overcoming these problems with the vision input, and at some point in the future, such problems could become minor.

4.2 Problems with the friction model

The first experiment was performed in order to try to tune the friction model to simulate the most common musical instrument driven by friction, i.e., a bow interacting with a string. It was confirmed that attempting to tune the physical violin model 'from scratch' is quite problematic, as often for certain values it fails to produce an audibly pleasant result. The same can be extended to applying independent control signals to change the violin model parameters - as the change of these variables in reality may be coupled (for instance, change of bow force may be coupled to change of bow velocity - as pointed out by both researh [13], and expert user evaluation).

So, regarding the issue of finding auditorily pleasant parameter values of the violin bow model - it may be a better approach to acquire recordings of control signals from a real-life violin first, which are certain to drive the physical model in a predictable range of attractive sonic output; and then use these as a base for further development of both a violin object, and for an independent bow controller object for a Reactable (or any kind of interface that might be applied to a violin physical model). For example, results described in [14] show that a precise augmented bow interface is an ideal controller for a bowed string physical model.

5. EXPERT USER EVALUATION

The comments collected during the informal expert user interview can be summarized as follows. Usage of a friction model in the Reactable could be interesting, but, before that, some improvements to the sound quality and control of the friction model are needed. The concept of dual objects is found interesting, beyond the notion of a controller - it has been suggested that an exciter object is made, that could similarly change parameters of any Reactable object. The concept of a single, movement-driven, friction object has not been found particularly interesting, and it does not necessarily require a physical model as an underlying sound engine.

So, in spite of some technical problems experienced during prototyping, there are hints that, provided the technical problems are overcome - objects that provide sonification of block movement in the context of the Reactable system, could potentially be a usable musical expression tool.

6. ACKNOWLEDGMENTS

The first author would like to thank researchers at the Music Technology group at UPF for welcoming him for the STSM and for providing useful input in the development of the application. The STSMs were sponsored by ConGAS, European Cost action 287 and SID, European Cost action 601 respectively.

7. REFERENCES

[1] A. Akay. Acoustics of friction. *The Journal of the Acoustical Society of America*, 111:1525, 2002.

[2] M. Alonso. Scientific Report from ConGAS Short Term Scientific Mission (STSM) to Stockholm. Technical report, Technical report, ConGAS Cost action 287, http://www. cost287. org/documentation/stsms, October 2006.

[3] S. Dimitrov. Scientific Report from ConGAS Short Term Scientific Mission (STSM) to Stockholm. Technical report, Technical report, ConGAS Cost action 287, http://www. cost287. org/documentation/stsms, March 2007.

[4] S. Dimitrov. Stsm development log - barcelona 2008, http://media.aau.dk/~sd/barca08/barcelona_stsm_08.html. World Wide Web electronic publication.

[5] K. Hansen. The Basics of Scratching. *Journal of New Music Research*, 31(4):357–365, 2002.

[6] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner. The reacTable: exploring the synergy between live music performance and tabletop

tangible interfaces. *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, 2007.

[7] S. Jordà, M. Kaltenbrunner, G. Geiger, and R. Bencina. The reacTable*. *Proceedings of the International Computer Music Conference (ICMC 2005), Barcelona, Spain*, 2005.

[8] M. Kaltenbrunner, S. Jordà, G. Geiger, and M. Alonso. The reacTable*: A Collaborative Musical Instrument. *Proc. of the TICE Workshop at the WETICE 2006*.

[9] E. Maestre, J. Bonada, M. Blaauw, A. Perez, and E. Guaus. ACQUISITION OF VIOLIN INSTRUMENTAL GESTURES USING A COMMERCIAL EMF TRACKING DEVICE.

[10] MTG-UPF. Reactable homepage, http://mtg.upf.es/reactable/. World Wide Web electronic publication.

[11] M. Puckette. Pure data: another integrated computer music environment. *Proc. the Second Intercollege Computer Music Concerts, Tachikawa*, pages 37–41, 1996.

[12] S. Serafin. Toward a generalized friction controller: from the bowed string to unusual musical instruments. *Proceedings of the 2004 conference on New interfaces for musical expression*, pages 108–111, 2004.

[13] S. Serafin, F. Avanzini, and D. Rocchesso. Bowed string simulation using an elasto-plastic friction model. *Proc. Stockholm Music Acoustics Conf.(SMAC 2003)*, pages 95–98.

[14] S. Serafin and D. Young. Bowed string physical model validation through use of a bow controller and examination of bow strokes. *Proc. Stockholm Musical Acoustics Meeting (SMAC)*, 2003.