Physical models and musical controllers – designing a novel electronic percussion instrument.

Katarzyna Chuchacz
Sonic Arts Research Centre
Queen's University Belfast
Belfast, Northern Ireland
+44 (0)28 9027 4761
kchuchacz01@qub.ac.uk

Sile O'Modhrain
Sonic Arts Research Centre
Queen's University Belfast,
Belfast, Northern Ireland
+44 (0)28 9097 4641
sile@qub.ac.uk

Roger Woods
Sonic Arts Research Centre
Queen's University Belfast,
Belfast, Northern Ireland
44 (0)28 9097 4081
r.woods@qub.ac.uk

ABSTRACT
A novel electronic percussion synthesizer prototype is presented. Our ambition is to design an instrument that will produce a high quality, realistic sound based on a physical modelling sound synthesis algorithm. This is achieved using a real-time Field Programmable Gate Array (FPGA) implementation of the model coupled to an interface that aims to make efficient use of all the subtle nuanced gestures of the instrumentalist. It is based on a complex physical model of the vibrating plate - the source of sound in the majority of percussion instruments. A Xilinx Virtex II pro FPGA core handles the sound synthesis computations with an 8 billion operations per second performance and has been designed in such a way to allow a high level of control and flexibility. Strategies are also presented to that allow the parametric space of the model to be mapped to the playing gestures of the percussionist.

Keywords
Physical Model, Electronic Percussion Instrument, FPGA.

1. INTRODUCTION
Over centuries successive generations of musical instrument designers have been developing new types of instruments and improving the level of expression and virtuosity of existing ones. Musicians are constantly looking for new creative ways of playing instruments. Advanced technology provides us with an opportunity to experiment with sound worlds outside the physical constraints of mechanical instruments and realize instruments with capabilities beyond the reach of acoustic instrument builders. Therefore the aspirations of contemporary instrument designers are not only focused on creating electronic replications of acoustic instruments but also proposing entirely new, creative, musical interfaces and building novel, inspiring instruments producing the sounds that could have never been produced by the acoustic system. In this paper, we focus on one class of such instruments, those that are percussive in nature.

There are numerous examples of electronic musical devices that were created to enhance or emulate acoustic percussion instruments such as the Buchla Thunder [1] which is a specialized MIDI controller. It senses a performer's touch and its pressure on 36 zones of playing surface, some of which respond to position of fingers as well. Another example is the Korg Wavedrum [2], a physics-based instrument that controls drum membrane vibration real-time simulation through acoustic sensing of hand and mallet strokes, giving more realistic results. ETabla [3, 4] exemplifies an instrument that has been created on the basis of careful observation and analysis of playing gestures and techniques of a performer to emulate real tabla - Indian traditional instrument. These are characterised by a number of constraints. Firstly, the majority are limited in terms of the size of the emulated instrument as realistic sounds of acoustic instruments e.g. gongs, timpanis, are usually unreachable. Their performance is also usually restricted to one, or a narrow group of, specific instruments. Another limitation is the sound synthesis algorithm complexity as instruments, e.g. gongs, tam-tams and cymbals create sounds whose timbre depends crucially on nonlinear elastic vibration effects [5] which existing electronic percussion instruments not only fail to reproduce but also reduce the ability to produce the realistic sounds of them. Finally, there are many different methods of capturing the gestural information of percussive strikes [6] but existing electronic percussion interfaces are mostly insensitive to the fine-grain nuances of the instrumentalists' playing techniques. Thus, a majority of the subtle details of articulation are out of reach and indeed are the most severe criticism when such systems are compared to conventional acoustic instruments.

The ambition to design an instrument free of the above limitations suggests adopting physical models as the sound synthesis algorithm. This offers a far wider range of rich sounds by providing, for example, models of different resonators and also an opportunity to create multi-channel sounds with internal coherence i.e. the various channels all coming from a single physical model. It further provides an opportunity to flexibly map playing gestures to the parameters space that control the state of the model. Here an extensive amount of research into the area of parameter mapping strategy importance in the design of musical electronic instruments has to be taken into consideration [7, 8, 9]. In [9] it was proven that in comparison to simple one – to – one mapping, more complex strategies that contain an additional abstract parameter layer and where the parameters are crosscoupled highly affects the instrument responsiveness to the performer actions. Their experiments also proved another characteristic that from the performer point of view makes the instrument to be more natural and engaging. This is the constant injection of instrumentalist's energy into the system. All these issues have to be taken into consideration in our musical controller design.

2. OBJECTIVES

The objectives of this present project can be grouped into three areas. Firstly, the real-time operation is crucial in securing the desired level of expressivity which involves a real-time strategy for implementing the physical modelling based sound synthesis algorithm. Secondly, the design of an interface to effectively capture the playing gestures of a performer specifically the expectations of professional instrumentalists, allowing them to extensively use their highly developed playing techniques and to provide a level of expressiveness and virtuosity, comparable to that of acoustic instruments. Finally, a clear strategy for parameter mapping between the hardware implementation and the controller has to be devised, in order to achieve such a high degree of expressiveness and virtuosity and raise the level of skill that can be supported by the instrument.

2.1 Sound synthesis algorithm

Our sound production mechanism, based on a classical Kirchhoff plate model, is described by partial differential equations (PDEs). Whilst varying numerical methods can be applied to solve PDEs iteratively on a computer, Finite Difference (FD) schemes were deemed to be the most obvious approach [10, 11] as they provide better modelling of non-linear factors such as transient pitch glides or build-up of high-frequency energy. The algorithm involves discretization of time and space to transform the PDEs to difference equations that can be then implemented digitally. It results in a grid of discrete points representing the transverse plate deflection approximation in both time and space coordinates. The value of each point of the grid is updated on the basis of its neighbours' values calculated in the previous iteration steps and its excitation value (if available). Figure 1 gives a grid fragment showing the sample update point and its significant neighbours in two following iteration steps.

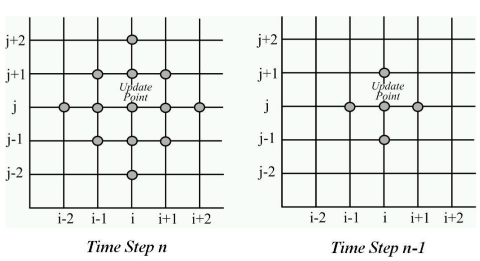

Figure 1. Update Point within the grid.

2.2 Hardware Implementation

Whilst FD schemes are a solid approach giving remarkably realistic results and satisfying the specific project requirements, they are computationally demanding and cannot be implemented on a single computer in real-time. For example a MatLab model running on a P4 Centrino 1.6 GHz PC with 512MB RAM takes over 35 minutes to produce 1 second of sound for a 100x100 square grid. As the algorithm is highly concurrent, a dedicated hardware solution based on FPGA is highly suitable as it is programmable and the built-in dedicated signal processing units allow high performance FD scheme implementations. In addition, the high level of memory access bandwidth allows several efficient strategies to be applied. Earlier work [12] showed it is possible to perform the calculations for sound synthesis faster than real-time. FPGAs are also desirable in terms of the project specification as they allow interfacing to a wide range of sensors. This feature has already been successfully exploited in the design of musical interfaces [13, 14].

3. DESIGN APPROACH

In the majority of electronic musical instruments, the overall design as well as the parameter mapping strategy results directly from the instrument interface capabilities. This means that the starting point for an instrument design is usually the method of interaction between the device and the performer. This contrasts with our approach where we exploit the fact that our instrument is based on the specific sound synthesis algorithm and start the design process from this basis. Given the FD implementation, that can be driven and read in a number of ways, this provides the possibility of deriving a highly flexible instrument where many parameters are fully open. This allows the exciting opportunity of connecting the professional player to the sound world of the model through the parameters' space before we actually define the instrument's controller itself.

4. PROTOTYPE

To date, our work has been focused on creating the hardware prototype of the synthesis model, presented on the Figure 2. The key element is the commercial hardware platform VMETRO VPF1 board that contains the Xilinx Virtex II Pro FPGA chip which implements the sound synthesis algorithm. A PowerPC microcontroller handles the communication with the host computer transferring the control data to the FPGA chip and retrieving synthesis output from the FPGA and transferring it to the host as a stream of samples. Communication is performed by the Ethernet connection that is set up between VPF1 and the host. The host PC is responsible for pre-processing control data and transferring them to the target. In the backward path the host retrieves the sample stream from the VPF1 board, processes them and outputs them to the sound card.

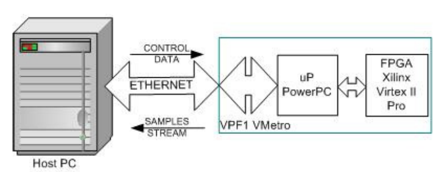

Figure 2. Prototype Architecture.

The Finite Difference scheme sound synthesis algorithm is implemented on the FPGA device as a network of processing elements (PEs) simultaneously performing calculations, resulting in the update of the grid points' values in every single iteration step. Each processing element is assigned to a sub-domain of grid points allowing a 100x100 grid to be implemented as a network of 10 PEs, each operating at 1000 of points and performing 0.8 billion operations per second. The PEs network controller implemented within the FPGA chip communicates with the rest of the system to receive the excitation and output the results.

One of the main purposes for implementing the instrument prototype was to distinguish the set of sound synthesis hardware implementation parameters and the parameter space that we are providing access to in real time. The parameters are presented in Table 1 together with their typical physical values and include grid point excitation, plate stiffness, linear damping, frequency dependent damping, grid size, plate size and sampling frequency.

Table 1. Sound synthesis parameter typical physical values.

	plate stiffness	freq. dep. damping	linear damping	sampling freq.
Parameter	K	b1	σ	Sf
Typ. value	15.26	0.005	0.98 1/s	44100 Hz

Grid point excitation can be applied in the form of single value or the function over the sub-domain of the grid points. Grid size represents the grid resolution i.e. number of grid points for each co-ordinate i.e. *Nx x Ny* (at the maximum of 100 x 100). Plate size represents the plate measurements (*Lx x Ly*). The FD scheme combines all these parameters into mathematical formulas which results in 5 abstract coefficients controlling the computations' hardware. This forms the bottom layer of the instrument parameter mapping structure as presented in Figure 3.

From the host computer site, the following accessible parameters can be accessed: an excitation value, address of the grid point to be excited, address of a grid point to be output, frequency dependent damping, linear damping, plate stiffness, grid size, plate size and sampling frequency. The last three are the parameters that are applied at the initialization stage of the model operation. The rest of them drive the model in real time.

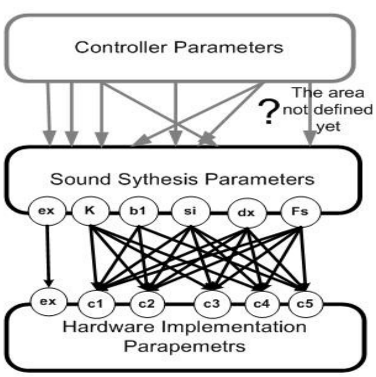

Figure 3. Parameters mapping structure.

The condition for model operation that affects the potential parameter space is the model stability condition. It directly determines the dependence of the grid spacing parameter (i.e. space between grid points $dx = Lx/Nx$) on stiffness parameter, sampling frequency and frequency dependent damping [10]. In our implementation, the spacing parameter *dx* is initialised at the beginning of the model operation so the stability condition affects the space of available stiffness parameter values.

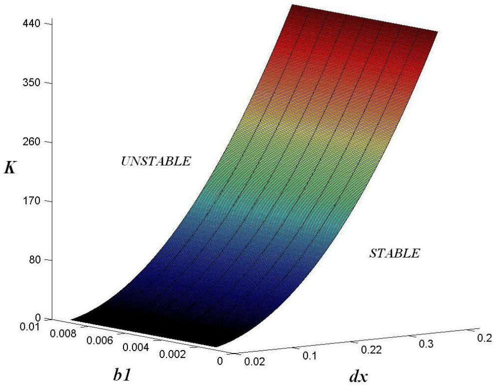

Figure 4. Stiffness parameter K availability space, resulting from the stability condition.

Figure 4 represents the stiffness stability boundary value, *K* over the spacing *dx* and frequency dependent damping *b*1 parameters. The sampling frequency *Fs* is constant (44100Hz). The surface of the plot splits up the space of the available *K* parameters into two areas. The area above the plot surface is beyond the stability and the bottom of the space and beneath the plot surface (including the plot surface) represents the area of *K* parameter within which the model is stable. As we can see parameter *b*1 does not have the significant influence on the value of stiffness comparing to *dx* parameter that is the main factor of the relationship.

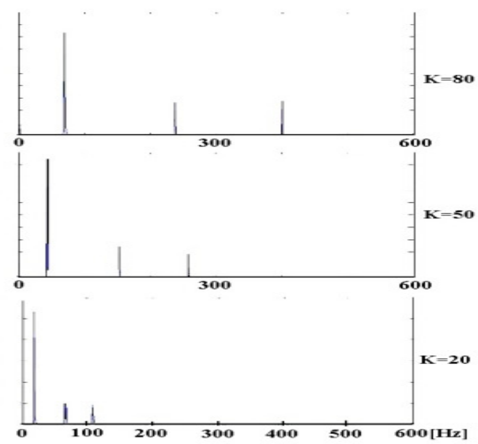

Figure 5. Spectrum of the produced sound for three different parameters K: 80, 50 and 20

As for the musical meaning of the parameters: the excitation value parameter represents the power of strike so it controls the volume of the attack; the sampling frequency *Fs,* influences sound quality; the linear damping parameter σ controls the time after which the sound falls with 30 dB; the frequency dependent damping controls high rates of loss at high frequencies; and the stiffness parameter controls the timbre and the pitch i.e. fundamental frequency. Figure 5 presents the comparison of the spectrum of the model response with different K values and shows that the lower stiffness parameter value results in a lower fundamental frequency. The rest of the parameters are set to values in Table 1.

Currently, the prototype implementation is limited in a number of ways. There is no capability of multi-node excitation i.e. it is not possible to excite and read multiples grid points simultaneously. An opportunity to excite modelled resonator with the digital signal other than single impulse is also not exploited within the prototype. All these will be incorporated within the final design.

5. FUTURE WORK

5.1 Hardware Implementation Issues

Building an application specific architecture is the main attraction using FPGAs as they offer considerable speed up. This is largely true for fixed point designs but the moderate floating point performance (5-10 GFLOPS/FPGA) is disappointing. Thus, it tends to be word length rather than processing that is the determining factor which must be given by the dynamic range needed by the instrumentalist. Prototype evaluation with professional musicians will help determine model constraints and the level of real-time interaction to meet musicians' expectations.

5.2 Motion Capture

In parallel with the synthesis module prototype design, we have already started detailed observation of gestures used by skilled percussion player in a real-world performance context. Within the first sessions, video capture of a range of playing techniques that a musician uses in order to achieve the desired instrument response, has been performed with the aim of using the Qualisys motion capture system to collect the quantitative motion data in future sessions. This will allow determination of gestural parameters and capture of the control of fundamental musical parameters such as volume, timbre, etc. This will decide the kinds of sensors needed for the instrument interface and will help define the temporal and spatial constraints of the playing gestures that need to be supported. Our ambition is to propose an interface that would come up to professional performers expectations and that would fully exploit the range of musically meaningful movements they wish to draw on in performance.

5.3 Mapping strategy

The final stage will focus on the design of the parameter mapping strategy to couple the players' movements to the model parameter space. This will involve working closely with percussion players to determine the flexibility that they would wish to retain in the instrument. More crucially, we hope to determine strategies for correlating model parameters to be controlled by a single playing gesture and thereby provide higher-level access to the capabilities of the model and in turn, specify the top layer of the parameter mapping graph in Figure 3. An important aspect of this phase of the project will be the development of the design approach, because we seek to provide the percussionists with whom we are working, with access to the model before we determine the interface. Our goal is to have player identify the elements of the model's sound world they wish to have access to and to determine how they wish to control these elements. From this process we will determine the gesture space that the interface of the instrument needs to support. Finally we will build the interface itself. The entire instrument will then be evaluated by a different group of percussionists. In this way we hope to arrive at an instrument whose gestural vocabulary and sound world are determined not by the technology we are using but by the kinds of actions suggested by the sound world of the model itself.

6. CONCLUSION

In this paper, the preliminary design steps that have been taken in creating a plate-based percussion synthesiser have been outlined. Emphasis is on creating an electronic percussion synthesizer capable of producing a high quality, realistic sound which makes efficient use of a wide range of the nuanced gestures of a skilled percussion player, giving them a level of expressivity comparable to acoustic percussion instruments.

7. REFERENCES

[1] Rich, R. Buchla Thunder. *Electronic Musician,* Aug. 1990.

[2] Rule, G. Keyboard reports: Korg wavedrum. *Keyboard,* 21(3), Mar. 1995.

[3] Kapur, A., Essl, G., Davidson, P., and Cook P. R. The Electronic Tabla Controller. *Journal of New Music Research*, 32(4) pp. 351-360, 2003.

[4] Kapur A., Davidson, P., Cook, P. R., Driessen, P. F., and Schloss, W. A. *Digitizing North Indian Performance*. Dep. of Elect. and Comp. Eng and Music, Uni. of Victoria, 2004.

[5] Rossing, T., Moore, F., and Wheeler, P. *The Science of Sound*, 3rd ed. Addison-Wesley, San Francisco, CA, 2002.

[6] Tindale A. R., et al., A Comparison of Sensor Strategies for Capturing Percussive Gestures. In *Proc. of Intl Conf. on New Interfaces for Musical Expression (NIME 05),* Vancouver, Canada, June 2005.

[7] Hunt A., Wanderley, M. M., and Kirk, R. Towards a Model for Instrumental Mapping in Expert Musical Interaction. In *Proc. of Intl Comp. Music Conf. (ICMC 2000)*. San Francisco, USA, 2000.

[8] Wanderley, M. M. Mapping Strategies for Real-time Computer Music. Special Issue. *Organised Sound 7(2),* Aug. 2002.

[9] Hunt A., Wanderley, M. M., and Paradis, M. The Importance Parameter Mapping in Electronic Instrument Design. *Journal of New Music Research*, Vol.32, No. 4, pp. 429-440, 2003.

[10] Bilbao, S. A Finite Difference Plate Model. In *Proc. of Intl Comp. Music Conf. (ICMC 2005)*. Barcelona, Spain, 2005.

[11] Bilbao, S. Sound Synthesis for Nonlinear Plates. In *Proc. of 8th Int. Conf. on Digital Audio Effects (DAFx'05).* Madrid, Spain, Sep. 2005.

[12] Motuk, E., Woods, R., and Bilbao, S. FPGA-Based Hardware for Physical Modelling Sound Synthesis by Finite Difference Schemes. In *Proc. of Int'l Conf. on Field Programmable Technology.* Singapore, Dec. 2005.

[13] Kartadina, S., The Gluion advantages of an FPGA-based sensor interface. In *Proc. of Intl Conf. on New Interfaces for Musical Expression (NIME 06),* Paris, France, Jun. 2006.

[14] Freed, A., Avizienis R., and Wright, M. Beyond 0-5V Expanding Sensor Integration Architectures. In *Proc. of Intl Conf. on New Interfaces for Musical Expression (NIME 06),* Paris, France, Jun. 2006.