KETTLE: A REAL-TIME MODEL FOR ORCHESTRAL TIMPANI

Panagiotis Papiotis
Music Technology Group,
Universitat Pompeu Fabra
panos.papiotis@gmail.com

Georgios Papaioannou
Department of Informatics,
Athens University of Economics and Business
gepap@aueb.gr

ABSTRACT

The orchestral timpani are a key component in western classical music, although their weight, size, and fragility make their transportation very difficult. Current commercial software synthesizers for the Orchestral Timpani are primarily sample-based and work with a MIDI keyboard, giving the user control over the note amplitude and pitch. This approach implements a virtual five-piece set of orchestral timpani, which is controlled using a pressure-sensitive graphics tablet. A brief analysis of the mechanics and playing techniques of the Timpani is presented, followed by their approximation by this model's control scheme and sound engine. Thereon, the details of the model's implementation are explained, and finally the results of the model are presented along with conclusions on the subject.

1. INTRODUCTION

The Timpani (also known as *kettledrums*) are a type of drum and, as such, belong to the musical family of percussion instruments; they consist of a thin membrane or *head* stretched over a large bowl traditionally made of copper. Unlike most drums, the timpani are capable of producing a strong pitch sensation when struck, and can be tuned via a mechanism that adjusts the tension of the membrane over the bowl. The most commonly used type of timpani today is the *pedal timpani*, in which the player controls a pedal with his/her foot which either increases or decreases the tension of the membrane, thus altering the drum's pitch. Older variations of timpani (e.g. the timpani used in the baroque period) are tuned by adjusting the tuning bolts individually.

Timpani are primarily used in sets of two or more, according to the music piece's requirements and, of course, the number of timpani available (it is not unusual for percussionists to play with one less drum than the piece requires, due to limitations). The largest of the commonly used timpani sets consists of five timpani, each with a diameter of roughly 32, 29, 26, 23 and 20 inches respectively.

Following the advances in software synthesizers, several commercially available products included sample-based models for the Orchestral Timpani as part of a larger Virtual Instrument Suite (examples include Edirol's *HQ-Orchestral*, IK Multimedia's *Miroslav Philharmonik* and Eastwest/Quantum Leap's *Advanced Orchestra Set*). These models, while useful for composers that wish to add to their compositions a few sporadic notes or pre-recorded timpani rolls (fast consecutive hits that create a rumbling sound), are designed to be controlled by a MIDI keyboard and thus do not take advantage of the instrument's continuous pitch range nor its vibrational mode-dependent timbre.

Our approach implements a real-time system that aims for two goals: to give the user control over every important factor that shapes the final sound of the instrument, while keeping the control scheme as simple and intuitive to the user as possible. A third and equally important goal was for the system to be easily available to potential users without the need for advanced control interfaces or high-performance computer systems.

The remainder of this paper is organized as follows: In Section 2, previous work on the subject is presented. Section 3 provides a brief overview of the instrument's unique sound mechanics, as well as the specific techniques employed by professional timpanists in order to achieve different sounds. Section 4 presents the design and architecture of our model. Section 5 contains the final model implementation details. Section 6 presents our results. Finally, Section 7 contains our conclusions as well as recommendations for future work.

2. PREVIOUS WORK

An important amount of research has gone into modeling the sound of percussion instruments and their abstractions (such as ideal membranes, plates etc.). One of the most promising and mature fields is *Physical Modeling* [1], or the simulation of the sound generation mechanism for a particular instrument. The clear advantage of this approach is that, through Physical Modeling, most parameters that partake in the creation of the sound are adjustable, thus making meaningful interaction with the model very intuitive for the user.

In this field, a very prominent approach is the use of the *Digital Waveguide Mesh* [2], in which a membrane is simulated by a set of one-dimensional digital waveguides, very much like a tennis racket's strings simulate an elastic membrane. Two-dimensional and three-dimensional realizations efficiently simulate membranes and acoustic spaces, while a significant improvement over this approach is *frequency warping* [3] in an interpolated three-dimensional mesh. Besides these cases, recent work on modeling rigid bars and plates [4] is also promising. An especially interesting combination of physical modeling for percussion instruments and multi-dimensional user input can be found in [5]. In this approach, a two-dimensional force matrix is used to both excite and damp a waveguide mesh, attempting to simulate the expressive qualities of hand drumming.

Another approach for two-dimensional membranes is presented in [6], making use of the so-called *Functional Transfer Methods* [7] - utilizing an analytical form of the modal parameters for a multi-dimensional differential system such as a membrane. There are more approaches to physically modeling nonlinear circular membranes [8], as well as two-membrane air coupling systems with strings such as a snare drum [9]. However, these approaches are relatively far from the specific physiology of the Timpani, and will not be discussed further.

There exist two complete numerical models specifically targeting the Timpani [10,11], and experimental results are promising. In [11], the model delves into the complexities of the Timpani as a physical system with great detail, taking into account the motion of the mallet and its nonlinear interaction with the membrane, the transverse displacement of the membrane, and the sound pressure inside and outside the enclosed cavity of the resonator bowl. Unfortunately, the aforementioned complexity of these models makes them unsuitable for a real-time system.

Finally, a very interesting and similar to this work project can be seen in [12], where a hardware sample-based model attempts to simulate the sound and control scheme of the kettledrum. This project is based on a custom-created drum pad with piezoelectric transducers, and sends input data from an electric circuit board via the RS-232 protocol to a PC where the audio output is calculated and played back. This project features a single kettledrum, one type of mallets, and uses an analog foot pedal to tune the drum.

3. TIMPANI SOUND PROPERTIES AND PLAYING TECHNIQUES

In order to provide a context for the subject at hand, key principles and terms related to the sound and playing conventions of the Timpani will be briefly presented. This is essential as this will be the base on which our model is designed.

3.1 Sound properties of the Timpani

As it was mentioned in the introduction, what distinguishes timpani from other drum-type instruments is the fact that they are able of producing a strong pitch sensation. The pitch is controlled using a tuning mechanism, and the tonal range of each drum is usually a perfect fifth (depending on the condition of the drum as well as its size). On certain types of Timpani, this range can be expanded to a full octave. Since the pitch is altered in a continuous manner by simply modifying the tension of the membrane, the Timpani are capable of producing any possible pitch within their tonal range, including microtonal intervals as well as “glissandi” or *sliding notes*.

Another important factor that sets Timpani apart from drums with a smaller skin surface, is the fact that they are highly sensitive to the distance of each hit from the center. Since circular membranes are two-dimensional, they can vibrate in many modes simultaneously and most of these modes are not harmonic; that is the frequencies of higher modes are not $n*f0, n=1, 2, \ldots, k$ (where f0 is the fundamental frequency and k the number of harmonics) as is the case with the harmonic series. Moreover, vibration in two dimensions means that there are two vibrational modes simultaneously and therefore two sets of nodal points; nodal circles and nodal diameters [13].

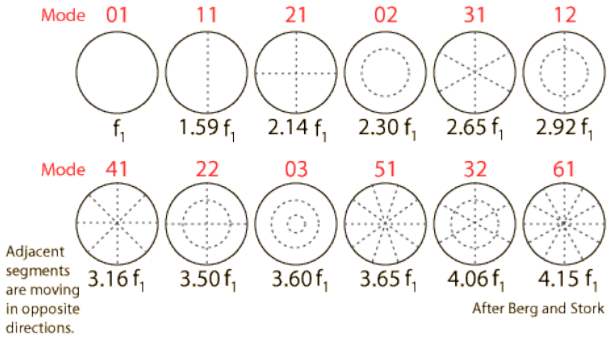


Figure 1. Vibrational Modes of a circular membrane.

When the (01) mode is excited, there are zero nodal diameters and one nodal circle on the boundaries of the membrane (see Fig.1). When the (11) mode is excited, there is one nodal diameter and one nodal circle, et cetera. However, not all of these modes produce a musically meaningful result; for this reason, there is a set of *preferred modes* [13], which timpanists and composers alike are very familiar with and exploit to achieve a different timbre that matches the musical occasion.

3.2 Playing techniques & conventions

As with most percussion instruments, the choice of mallets in Timpani performances is very important. There are several types of mallets which vary on the hardness of the mallet head and the material it is made of, as well as the elasticity of the mallet’s shaft. Almost every musical piece instructs the player on the type of mallets to use, ranging from soft felt mallets, flannel mallets, harder mallets used also for marimbaphones, and mallets with wooden heads. Each type changes not only the attack of each note, but also the timbre that is produced.

Most musical pieces written for/including timpani also demand that the timpanist tunes the drums as he/she plays. Almost all advanced pieces change the set’s tuning during the piece, and many require that the timpanist changes the tuning of a drum while it is still resonating.

This sliding note technique is called *glissando*, and is commonly used in conjunction with timpani rolls.

The concept of a drum roll is central to all percussion instruments, and it consists of fast consecutive strikes on the drum's surface that alternate between the player's left and right hand. This is especially important for the timpani, as it creates a rumbling, drone-like sound which is pivotal all timpani pieces.

Finally, a known technique in music for timpani is membrane *damping*. A timpanist often touches the membrane in order to mute that drum; another technique is the use of *mutes*, which resemble small pillows and are placed on the membrane throughout the musical piece while the timpanist plays, in order to reduce the vibration and "dry out" the sound produced.

4. DESIGN OF THE MODEL

In order to satisfy the model's requirements in terms of sound synthesis and user interaction scheme, these two attributes needed to take full advantage of each other. Therefore, in the following section we present our choice for the model's sound engine and control scheme, as well as the final design of the system.

4.1 Sound synthesis

We considered two different approaches for the sound synthesis engine: Physical Modeling, and Sample-based synthesis.

Judging by the quality of the existing physical and numerical models for timpani presented in section 2, our initial idea was to implement our model based on a physical modeling synthesis algorithm such as the digital waveguide mesh, coupled with a model for the drum's resonator bowl. Considering recent advances in the field, it is easy to deduce that an extremely realistic physical model would produce the best results; on the other hand, physical models include numerous mathematical calculations which, in a three-dimensional system such as the kettledrum, would not only require significant processing power but also potentially introduce numerical errors in order to compute the output in real time.

The second approach would be a sample-based model; such a system is implemented in the commercially available timpani models, and could be partially connected to the pathologies shared by them. However, one cannot separate the interface from the virtual instrument: MIDI keyboards only allow for discretized pitch and note amplitude based on the velocity and number of the key pressed. Such a system is effective for instruments with keys (such as pianos or organs), or even some string and wind instruments. If an appropriate control interface provides more input dimensions, the manner in which samples are interpolated to produce the synthesized sound can produce very convincing results [1]. Moreover, sample-based synthesis is far less computationally intensive than a physical model, which is very important in a real-time application. For these reasons, we decided to implement a sample-based model, which is controlled by an appropriately multi-dimensional interface.

4.2 Control scheme

As described in section 3, the most important parameters that shape the sound of the Timpani are the *hit coordinates*, the *type of mallets* used, the *tuning* of each drum, the use of *damping/mutes,* and of course the *strength* of the hit.

From the above parameters it is obvious that the interface used to control our model must possess at least two important qualities; provide two-dimensional coordinates for the position of the mallet's strike, and be pressure-sensitive. All other parameters (type of mallets, tuning, and mutes) can be selected using the existing controllers in a computer, such as the mouse and/or keyboard.

There are several interfaces that provide two-dimensional, pressure sensitive input. However, one of our goals was to design a system that would be available to the majority of potential users; thus, we had to exclude expensive controllers such as the hit coordinate-sensitive Mandala drum pad [14] or touch screens, as well as custom-designed interfaces such as pads equipped with multiple piezoelectric transducers.

Another alternative would be a webcam-based approach; unfortunately, ordinary webcams have a relatively low frame rate, and are prone to errors caused by distractions within the capture area.

As our system required a widely available but at the same time relatively precise interface, our choice was to use a digitizer tablet. Although tablets do not provide tactile feedback (such as the previous solutions), they are accessible due to their low starting price (~50 Euros), and provide high capture precision (albeit with a restricted surface area, especially for cheap models). Moreover, most digitizer tablets also feature scroll wheels and programmable buttons; these extra controls can be used to change the tuning of the timpani as well as the mallet type.

4.3 The model

Having decided to use a sample-based synthesis algorithm, we needed to determine the number and nature of the samples that would comprise the soundbank for our model.

4.3.1 Hit coordinates

In timpani sheet music, there are predominantly four different written notes to signify the coordinate of the hit (see Fig. 2).

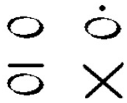


Figure 2. Sheet notation for hit coordinates.

From top to bottom and left to right, these symbols specify the hit position as *normal* (i.e. at ¾r from the drum's center, where *r* equals the radius of the drum), *staccato* or close to the drum's center, *legato* or close to the drum's rim, and the final symbol signifies a hit either on the center or on the rim of the drum's head, further explained using footnotes on the sheet music. Thus, five different hit positions were recorded in the soundbank.

4.3.2 Dynamics

In order to have a dynamically varied model, it is not sufficient to play back the samples at different levels of amplitude. A kettledrum's sound differs greatly when it is struck softly rather than strongly, as the harmonics and partials present in the attack and release parts of the note evolve differently. Therefore, it was pivotal to the model's realistic nature that several dynamic levels were recorded.

In classical music, dynamics range from *ppp* (pianissisimo, very very soft) to *fff* (fortissisimo, very very loud). Of all those dynamic values, the most commonly met are *pianissimo*, *piano*, *mezzo piano, mezzo forte, forte, and fortissimo*. Dynamics outside this range (as well as the *mezzo piano* dynamic) are meaningful musically, but can be achieved by adjusting the level of a recorded sample since their distance from the next closest value is small. In our model, the samples were recorded in five different dynamic values (pp, p, mf, f, ff).

4.3.3 Types of mallets

There are many different types of mallets that can be used in order to create a unique sound. Often timpanists freely choose the type of mallet to achieve a specific sound, even if it contradicts the piece's instructions. Overall, timpani mallets fall into three categories: *soft* mallets, with heads typically made from wool or flannel; *hard* mallets with wooden heads wrapped in thick thread resembling those used for chromatic percussion such as the Marimba or the Vibraphone; and finally *wooden* mallets, or mallets with wooden heads that are wrapped in leather.

In order to control the size of the soundbank, the samples were recorded with three different pairs of mallets, with each pair being the most commonly used in its category: soft wool mallets, Marimba mallets with thread-wrapped heads and bare wooden mallets.

4.3.4 Tuning range

Western music theory divides an octave in 12 intervals, the *chromatic scale*. Of these 12 intervals, only eight fit in the tonal range of a typical kettledrum. However, to record all 8 notes would not only result in an exceedingly large soundbank, but would also deprive the model of the ability to *slide* between notes, achieving microtonal intervals and performing glissandi. Thus, an efficient real-time algorithm had to be created so that each drum could achieve any possible tuning between its lowest and highest possible note.

The most obvious way to achieve such a result would be to simply pitch-shift a middle note, in order to emulate higher and lower pitches. However, it was evident early in out experiments that such a solution would neglect the difference between the harmonic series present in two notes with different pitch (see Fig.3).

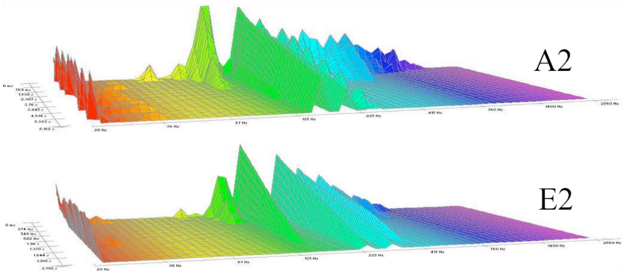


Figure 3. A2 and E2 played on a 29'' kettledrum.

The most important segments of a Timpani note are respectively its *attack* and *decay*, as is the case with most instruments. Figure 4 shows the spectrogram of a C2 note played on a 29'' kettledrum:

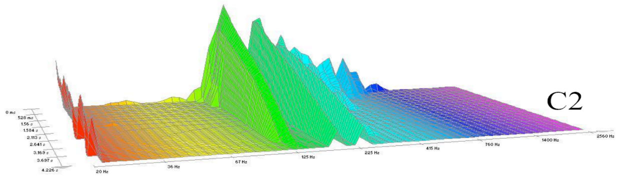


Figure 4. C2 played on a 29'' kettledrum.

The difference between Figure 3 and Figure 4 is evident: to pitch-shift either of these notes in order to match the other would distort the original timbre of the note we are trying to get. Figure 5 shows the spectrograms of two different C2 notes, produced by pitch-shifting an A2 and an E2 respectively:

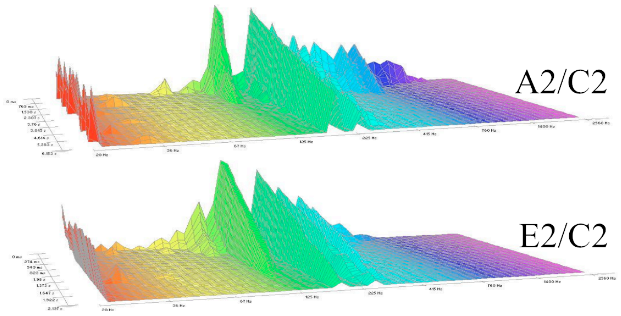
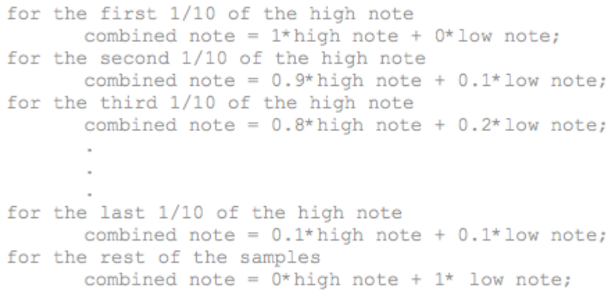


Figure 5. A2 and E2 pitch-shifted to reach C2.

It can be observed in Figure 5 that each shifted note has different similarities to the actual C2 note. Namely, the *attack* part of the pitched-down E2 note is a good approximation of the C2 note's attack, whereas the *decay* part of the pitched-up A2 note matches with the decay of the natural C note. Of course, this can be observed better acoustically, but a common element that is present both acoustically and visually is that, while the pitched-down note provides a more precise representation of the harmonics contained in the actual note, the pitched-up note retains a better approximation of the duration and time evolution of the original note (since the duration of the pitched-down E2 note is a mere 2.2 seconds, while the duration of the pitched-up A2 note is 6.2 seconds).

Taking advantage of this knowledge, we implemented a crossfading algorithm that gradually combines the two pitch-shifted notes to create the final note. This way, we managed to both restrict the size of our soundbank, and to achieve all possible tunings within each drum's tonal range.

Pseudocode for this crossfading algorithm can be seen in Figure 6. In essence, our engine utilizes two notes, each one at the boundary of the kettledrum's tonal range; these notes are aptly named the *high* and *low* note, respectively. Using time-domain pitch shifting, the *high* note is shifted downwards and the *low* note is shifted upwards to match the desired pitch.



```
for the first 1/10 of the high note
       combined note = 1*high note + 0*low note;
for the second 1/10 of the high note
       combined note = 0.9*high note + 0.1*low note;
for the third 1/10 of the high note
       combined note = 0.8*high note + 0.2*low note;
       .
       .
       .
for the last 1/10 of the high note
       combined note = 0.1*high note + 0.1*low note;
for the rest of the samples
       combined note = 0*high note + 1* low note;
```

Figure 6. Pseudocode for the crossfading algorithm.

One must keep in mind that the pitched-down *high* note is significantly shorter in duration than the pitched-up *low* note; the crossfading algorithm utilizes the attack of the *high* note and gradually makes the transition to the decay of the *low* note, thus emulating the duration of the desired note as well.

Note that all calculations are done in a single-sample basis, while the audio buffer is also 'fed' one sample at a time. With this technique, the user can modify the pitch of each drum in real time without any audible artifacts or abnormalities in the produced note.

4.3.5 Damping

It is very hard to simulate a damped note from an existing recording of a 'normal' note. Seeing as our soundbank was increasing in a rapid manner, we decided to omit the choice of damped notes or mutes for the time being. However, this is an important feature and methods to include it are currently under investigation.

5. IMPLEMENTATION OF THE MODEL

Our final Timpani model consists of a pressure-sensitive digitizer tablet as the surface for all five timpani, using the stylus as a mallet and the tablet's scroll wheels as tuning cogs. The model's soundbank consists of 750 recorded samples, that feature 5 dynamic values, 5 hit points, 3 mallet types, and 2 notes for each of the five different kettledrums.

5.1 Sample acquisition

All samples were recorded at a conservatorium, using a five-piece set of Adams symphonic timpani. In order to control the exact dynamic and hit position for each sample, a simple mechanism was constructed (see Fig. 7); the mechanism consisted of an elastic length of metal suspended over the drum's head, on which the mallet was attached.

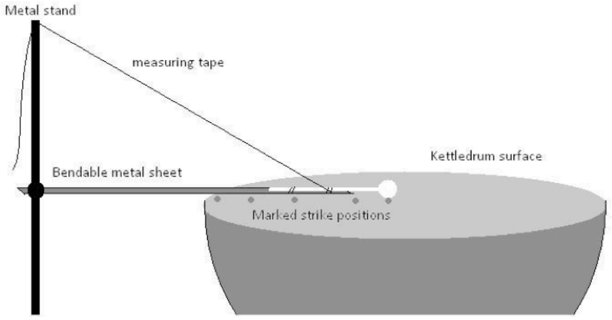


Figure 7. The mechanism created to control hit strength and position.

Each sample was given a unique ID number, in order to simplify the loading and management of samples. The ID was created with the following format: 1000*[type of mallet] - 0 for soft, 1 for hard and 2 for wooden mallets; 100*[number of drum] – 0 for the 20'' drum, 1 for the 23'' drum etc; 10*[dynamic value] – 0 for pianissimo, 2 for piano etc; and finally, 1*[hit position] – 0 for the center of the drum, 1 for staccato, and so on. For example, sample 2431 is a strike on the *staccato* position, with a *mezzoforte* dynamic value, on the *32-inch* drum with *hard* mallets.

5.2 Coding language & Audio processing API

The two main programming languages considered for the project implementation were Java and C++. Due to limited tablet support in Linux-based systems and relatively high latency in sample pre-mixing and event processing in Java, a Win-32 C++ implementation was chosen.

Our API of choice was the *Synthesis Toolkit* (STK) [15] in combination with *RtAudio* [16], due to its low-level functionality, versatility, and low latency performance. Moreover, STK and RtAudio are cross-platform; this allows for future porting of this project in order to make it available in Linux- and OSX-based systems.

5.3 Memory management

Currently, every sample is loaded in the memory during the program's startup; unfortunately, this way the application takes up to 1.100 megabytes of memory. However, it is a very easy task to modify the application so it only loads the samples needed for the mallet type chosen, thus reducing its memory footprint to 350 megabytes of RAM. Since the project intended to give the player full control over the timpani set as a timpanist would have, it was deemed important that the user could switch between mallets without any delay or additional loading.

5.4 User Interface

Hit coordinates and strength are retrieved through the input provided by the tablet. For visualization purposes, a printed sheet depicting the position of each drum is placed on the tablet, as seen in Figure 8.

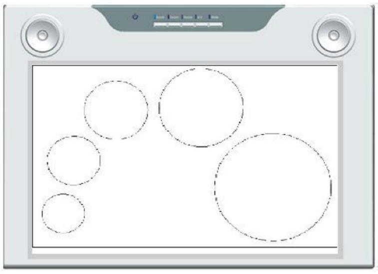


Figure 8. An example of the printed sheet used to visualize the position & size of each kettledrum.

A significant drawback of the control scheme is the fact that tablet drivers only allow for one stylus to be used. We bypassed this restriction by applying a simple solution; when the stylus hits the tablet, the first note is produced. For as long as the stylus stays pressed on the tablet, the x-axis direction in which it is moving is recorded; when the direction changes, a new note is produced. This way, the user can control the speed and dynamics of the timpani roll by ‘drawing’ smaller or wider circles on the tablet surface.

The system uses the default Windows libraries for tablet devices that ships bundled with all tablet drivers, thus eliminating the need for specific libraries. In the event that a tablet does not have scrolling wheels, the mouse wheel can be used to tune the each kettledrum in real time. Finally, the selection of mallet type is available to the user through the application’s GUI.

6. RESULTS

6.1 Synthesis quality

The spectrograms of three real and synthesized notes can be seen in Figures 9-11. The synthesized notes were produced by the tuning algorithm presented in 4.3.4, using A2 and E2 as the low and high note respectively. As it can be seen, the overall structure and evolution of the harmonic series is faithful to the recorded notes; however, in cases such as the D2 note it can be seen that the higher harmonics are slightly enhanced.

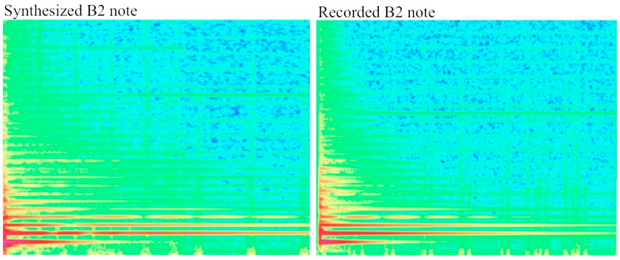


Figure 9. Comparison between a synthesized B2 note played on a 29’’ kettledrum, and the real note.

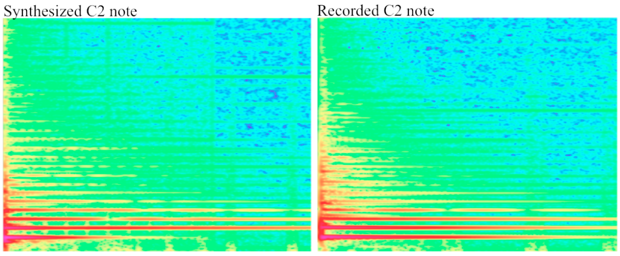


Figure 10. Comparison between a synthesized C2 note played on a 29’’ kettledrum, and the real note.

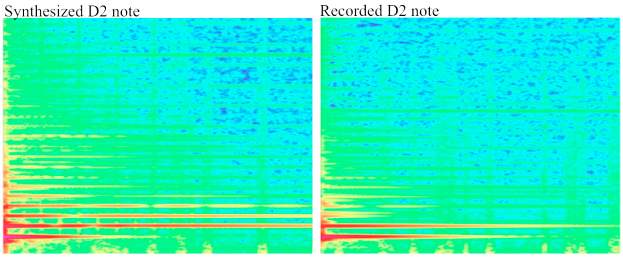


Figure 11. Comparison between a synthesized D2 note played on a 29’’ kettledrum, and the real note.

An audiovisual demonstration of the model will soon be publically available on the Internet for evaluation purposes, on the address provided in [17].

6.2 Audio Latency

The real-time response of the model yields an audio latency of 25 milliseconds when tested with a bulk Realtek on-board audio card using the DirectSound audio driver. For an external audio card using an ASIO driver, audio latency is below 12 milliseconds.

7. CONCLUSIONS & FUTURE WORK

In this paper, we presented a working system that implements a real-time model of the Orchestral Timpani, controlled by a digitizer tablet. The user has control over almost every parameter that defines and shapes the sound of the instrument, and these parameters can be modified and adjusted in real time. Our results demonstrate that we successfully emulate the sound of the Orchestral Timpani, while achieving low latency.

On the other hand, there are certain drawbacks to our approach, the first of which is its rather high memory demands. A compromise would be to only keep the samples for one type of mallets loaded in the memory, imposing a brief loading time when the user chooses another type of mallets. However, a more efficient memory management scheme could be utilized in order to constantly maintain all samples loaded in the memory.

Another drawback is the fact that tablets only allow for one stylus to be used. This has its toll on the interaction scheme, as the instrument was designed to work with two mallets. Moreover, tablets do not provide tactile feedback of any kind, a characteristic which is inextricably linked with the process of playing percussion instruments; to this end, multi-touch input interfaces that provide tactile feedback to the user must be investigated in the future.

Finally, the model used in this project applies to many percussion instruments sensitive to different hit coordinates & hit strengths such as cymbals, bass drums, and gongs. If a new soundbank was recorded to feature such instruments, almost all algorithms and methods created for this project could easily be modified to include such instruments.

8. REFERENCES

[1] J.O. Smith: "Virtual Acoustic Musical Instruments: Review and update", *Journal of New Music Research* 33(3), pp. 283-304, 2004.

[2] S.V. Duyne, J.O. Smith: "The 2-D Waveguide Mesh", *IEEE Workshop on In Applications of Signal Processing to Audio and Acoustics*, pp. 177-180, 1993.

[3] L. Savioja, V. Välimäki: "Reduction of the dispersion error in the triangular digital waveguide mesh using frequency warping," *IEEE Signal Processing Letters*, vol. 6, no. 3, pp. 58–60, 1999.

[4] S. Bilbao: "A modular percussion synthesis environment", *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, 2009

[5] R. Jones, A. Schloss: "Controlling a Physical Model with a 2D Force Matrix", *Conference on New Interfaces for Musical Expression*, New York, 2007.

[6] L. Trautmann, S. Petrausch, R. Rabenstetin: "Physical Modeling of drums by transfer function methods", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Vol. 5, pp. 3385-3388, 2001.

[7] L. Trautmann and R. Rabenstein: "Digital Sound Synthesis by Physical Modeling Using the Functional Transformation Method". New York: Kluwer Academic, 2003.

[8] R. Marogna, F. Avanzini: "Physically-based Synthesis of Nonlinear circular membranes", *Proc. of the 12th Int. Conference on Digital Audio Effects (DAFx-09)*, 2009

[9] F.Avanzini, R. Marogna: "A Modular Physically Based Approach to the Sound Synthesis of Membrane Percussion Instruments", *IEEE Transactions on Audio, Speech, and Language Processing*, 18(4), 2010

[10] R. E. Davis, "Mathematical modeling of the orchestral timpani", *Ph.D thesis*, Physics Department, Purdue University, 1988.

[11] P. Joly A. Chaigne, L. Rhaouti: "Time-domain modelling and numerical simulation of a kettledrum", *Journal of the Acoustic Society of America* 105(6), pp. 3545-3562, 1999.

[12] S. Emsen, C. Heal, R. Sunderland, R. Willis: "GDP 7 Electronic Timpani", University of Southampton - Faculty of Engineering and Applied Science, 2002.

[13] C. R. Nave: "Hyperphysics – The Timpani", http://hyperphysics.phy-astr.gsu.edu/hbase/music/timpani.html, retrieved on April 2010.

[14] Synesthesia Corp, "The Mandala Drum", http://www.synesthesiacorp.com/, retrieved on April 2010.

[15] P.R. Cook, G.P. Scavone: "The Synthesis Toolkit in C++", https://ccrma.stanford.edu/software/stk/, retrieved on April 2010.

[16] G.P. Scavone: "The RtAudio API", http://www.music.mcgill.ca/~gary/rtaudio/, retrieved on April 2010.

[17] http://www.youtube.com/user/kettleVirtualTimpani