HELSINKI UNIVERSITY OF TECHNOLOGY
Department of Electrical and Communications Engineering
Laboratory of Acoustics and Audio Signal Processing

Leevi Peltola

Analysis, Parametric Synthesis, and Control of Hand Clapping Sounds

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo, Dec 07, 2004

Supervisor:	Professor Vesa Välimäki
Instructor:	Dr. Cumhur Erkut

HELSINKI UNIVERSITY
OF TECHNOLOGY

ABSTRACT OF THE
MASTER'S THESIS

Author:	Leevi Peltola		
Name of the thesis:	Analysis, Parametric Synthesis, and Control of Hand Clapping Sounds		
Date:	Dec 07, 2004	**Number of pages:**	58
Department:	Electrical and Communications Engineering		
Professorship:	S-89		
Supervisor:	Prof. Vesa Välimäki		
Instructor:	Dr. Cumhur Erkut		

The purpose of this thesis was to create a synthesis and control model for hand clapping sounds. The first part of this thesis is a synthesis model for a sound of hand clapping. The synthesis model is based on a two-pole resonator filter whose coefficients are derived from measurements. The filter is excited with short noise pulses to create a simplified but realistic resynthesis of the sound of hand clapping. As the synthesis is based on measurements made in anechoic chamber, also an artificial room reverberation is included.

The second part of this thesis focuses on creating control models for the synthesis model. The simplest control model is an imitation of one clapper. The fluctuation of clapping rate is based on the analysis of recorded clapping sequences. More advanced control model addresses to the phenomenon of synchronized applause and is based on the Kuramoto model of coupled nonlinear oscillators. Every clapper in an audience is modeled as an oscillator that has its own preferred clapping rate. When a synchronization is turned on these oscillators start to change their clapping rate and phase to find a synchronization. Both synthesis and control model are implemented as a real-time software.

Keywords: digital audio signal processing, computer music, acoustic resonator filters, control model, Kuramoto model.

TEKNILLINEN KORKEAKOULU		DIPLOMITYÖN TIIVISTELMÄ
Tekijä:	Leevi Peltola	
Työn nimi:	Analysis, Parametric Synthesis, and Control of Hand Clapping Sounds	
Päivämäärä:	07.12.2004	**Sivuja:** 58
Osasto:	Sähkö- ja tietoliikennetekniikka	
Professuuri:	S-89	
Työn valvoja:	Prof. Vesa Välimäki	
Työn ohjaaja:	Dr. Cumhur Erkut	

Tämän työn tarkoituksena oli tehdä synteesimalli käsien taputukselle. Synteesimalli perustuu kaksinapaiseen resonaattorisuotimeen. Suotimen kertoimet on määritelty kaiuttomassa huoneessa tehdyistä testiäänityksistä. Suotimeen syötetään lyhyitä kohinapulsseja ja tuloksena saadaan yksinkertaistettu, mutta realistinen käsien taputuksen ääni. Realistisuuden lisäämiseksi tulokseen myös lisättiin keinotekoinen jälkikaiunta.

Toinen tavoite oli kehittää realistisia ohjausmalleja synteettiselle käsien taputukselle. Yksinkertaisin malli on imitaatio yhden ihmisen antamasta aplodista. Siinä taputustahtia vaihdellaan testinauhoituksista saatujen analyysien perusteella. Toinen hieman kehittyneempi ohjausmalli keskittyy ilmiöön, jossa yleisö synkronoi taputustahtinsa. Se perustuu Kuramoton malliin kytketyistä epälineaarisista oskillaattoreista. Jokainen ihminen yleisössä on mallinnettu oskillaattorilla, jolla on oma yksilöllinen taputustahtinsa. Kun synkronointi on kytketty päälle, oskillaattorit alkavat muuttaa taputuksen tahtia ja vaihetta, jotta yhteinen tahti löytyisi. Sekä synteesiettä ohjausmalli toteutettiin reaaliajassa toimivana ohjelmistona.

Avainsanat: Digitaalinen äänisignaalinkäsittely, tietokonemusiikki, akustinen resonaattorisuodin, ohjausmalli, Kuramoto-malli.

Acknowledgements

This Master's thesis has been done at the Laboratory of Acoustics and Audio Signal Processing at Helsinki University of Technology. The research was funded by the Academy of Finland CAPSAS-project (project no. 104934).

First I would like to thank my supervisor Vesa Välimäki and my instructor Cumhur Erkut for the possibility to work in this interesting project. Both of them have provided helpful comments and suggestions in both scientific and grammatical issues.

I would also like to thank my mom and my brother for the housing and Fatima for all the long distance support.

Otaniemi, December 7, 2004

Leevi Peltola

Contents

List of Abbreviations

CAPSAS	Control, Analysis, and Parametric Synthesis of Audio Signals
CPU	Central Processing Unit
FFT	Fast Fourier Transform
FM	Frequency Modulation
IIR	Infinite Impulse Response
LPC	Linear Predictive Coding
LPLE	Linear Prediction With Low-Frequency Emphasis
MIDI	Musical Instrument Digital Interface
OOI	Onset-to-Onset Interval
PhISEM	Physically Informed Stochastic Event Modeling
Pd	Pure Data
WLPC	Frequency-Warped Linear Predictive Coding

List of Notations

A_0	Gain factor
B	Bandwidth of a resonator
f_c	Center frequency of a resonator
g	Gain of a resonator
OOI_{NAT}	Onset-to-Onset Interval in natural clapping mode
OOI_{SYNC}	Onset-to-Onset Interval in synchronized clapping mode
R	Pole radius of a resonator
$x(n)$	Input of a resonator at time n
$y(n)$	Output of a resonator at time n
θ	Pole angle
ψ	Resonant frequency

Chapter 1

Introduction

This thesis deals with the parametric synthesis of percussive everyday sounds and above all the synthesis of hand clapping. Clapping of hands is very popular audible activity in every culture [39]. In Western countries its most common function is to show approval and favor. In many cultures and various kinds of folk music it is used as a rhythmic instrument. However, there have not been many studies about it. It can be analyzed as an applause of many people clapping hands together or as an individual sound generating process of two hands making a sound burst when hitting together.

In general sound synthesis is used by musicians and movie industry, as well as virtual reality applications and computer games. Synthetic hand clapping can be used for example in sport games when modeling the applause given by the audience or to make the applause in live recordings more intense. It can also be used as a feedback in virtual reality and computer games; user can be rewarded with an enthusiastic applause and negative feedback can be given with a bored applause. Applause is also one of General MIDI [27] sound effects and a hand clap is one of the percussion sounds of MIDI channel no. 10.

As the control of the synthesis is based on the physics of sound producing event it can be easily adjusted to produce different kinds of everyday sounds and there are numerous possible applications for it.

The goal of this work is to create physically-based synthesis of hand clapping with physically-based control parameters. If the control parameters are based on the physics of sound producing event, they can be easily tuned to model other similar sounds like rain drops or the sound of objects falling to a floor. On the other hand, the goal is not to make exact resynthesis of hand clapping but rather perceptionally sharp simplified cartoonification of it.

Another goal is to create physically-based and computationally efficient control model with which the synthesis of a single clap can be extended to a realistic hand clapping sequence or even an applause of a large amount of people.

1.1 Contents of This Thesis

The rest of this thesis is structured as follows. In Chapter 2 the literature related to subject is reviewed. First, several studies related to perception of sound source characteristics are described. Then general techniques of sound synthesis are reviewed and the physically-based synthesis of an individual sound event with some simplifications is described. Finally some works about the control and articulation of these sound events are described.

In Chapter 3 the measurements made for analysis of the characteristics of hand clapping are presented. Then the analysis results for estimating the modal parameters of a single clap are described and finally the cartoonificated resynthesis model of a single clap is presented. Chapter 4 will explain the control model of a single clapper producing a sequence hand clapping. The model is based on the statistics from measurements and previous studies. Then a model for a larger group of clappers is presented in Chapter 5. The theory of coupled nonlinear oscillators is used in the model for rhythmic applause. Finally in Chapter 6 the techniques presented in this thesis are summarized and the conclusions of the work are presented. Also the direction for possible future studies are pointed.

Chapter 2

Previous Research on Everyday Sounds and Synthesis

2.1 Acoustic Source Characteristics

2.1.1 Sound of Two Hands Clapping

In 1987, Repp [39] investigated the sound of two hands clapping by means of acoustical analyses and perceptual experiments. He studied the sound of clapping as an individual articulatory activity as claps may convey information about hand size and configuration.

Repp used in his tests 20 subjects who were asked to clap at their normal rate for 10 seconds in a sound insulated room. Notes of the subjects' hand size and configuration were taken and the spectra was calculated. The spectral shapes of individual clap tones were varying remarkably, but there was no evidence of the influence of sex or hand size on the clap spectrum. He also measured his own clapping with eight different clapping modes (See Fig. 2.1). The principal component analysis made for clapping spectral shapes consisted of four significant factors and two of these factors were correlated with the position of hands. If the hands were positioned so that the palms struck each other there was a narrow frequency peak below 1 kHz together with a notch around 2.5 kHz. If the hands were clapped so that the fingers of one hand struck the palm of other hand there was broad spectral peak in the vicinity of 2 kHz. Conclusion was that low-frequency peak represents the palm-to-palm resonance and the mid-frequency peak represents the fingers to palm resonance.

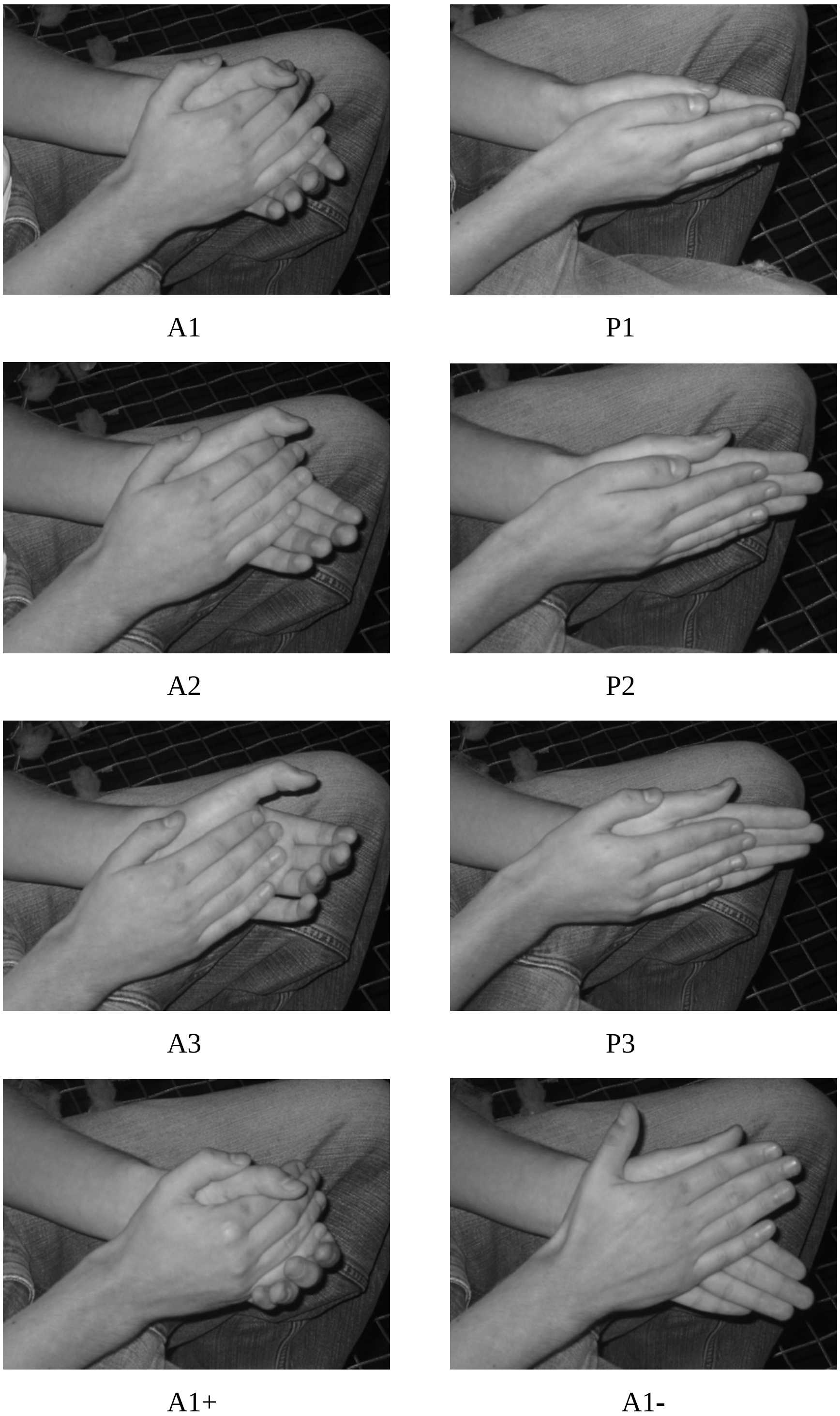

Figure 2.1: Different hand clapping modes after Repp [39]

.

After that, the perception of previous measurements was studied. The purpose of the study was to determine whether the 20 subjects were able to extract information of clappers' sex and hand size. Also the recognition of the subject's own hand clapping was studied. Of the guesses whether the clapper was male or female only 54% were correct. The response trends were stereotypical so that males were expected to clap slow, loud and low-pitched while females were expected to clap fast, soft and high-pitched. However, this was not the case and differences were caused mainly by hand configurations rather than hand size. 46% of the subjects were able to recognize themselves. As the hand size showed no effects on clap spectrum, also the perception of hand configuration was investigated. The subjects could generally recognize the clapping styles with palm-to-palm resonance peak below 1 kHz. However, the judgments were also based on clapping rates and other factors that were not related to the clapping style.

Repp's study [39] was the first exploration of this little studied subject and some methodological shortcomings were present. The recording procedure was not optimal; measurements should be done in an anechoic chamber and the distance from the microphone should be controlled more carefully. The spectral analysis was made on low-pass filtered signals and possible information above 5 kHz was lost.

Tousman and his collagues [49] made some additional experiments on Repp's results. The purpose of this experiment was to more directly assess the capabilities of untrained listeners to determine source characteristics, and to determine the physical aspects of the acoustic signal that are highly correlated with the source. He also tried to train subjects so that the performance could be improved. Results indicated that untrained listeners were able to distinguish hand position from only a few clappers. Even the training over 360 trials did not significantly improve results.

2.1.2 Walking Sounds

Li and his collagues [32] investigated the ability of subjects to perceive the gender by the sound of walking. A number of acoustical and physical properties were measured and subjected to statistical analysis to identify the properties that differentiate male and female walking. The perception of walker gender is important in demonstrating that listeners can identify some source characteristics from the properties of the acoustical signal, even though Repp could not show that it is possible in the case of clapping hands.

In this investigation eight males and eight females walked normally on a hardwood stage using same kind of shoes. Recordings that consisted of four footsteps were presented to listeners over the headphones. Males were identified correctly on 69% of the trials and females were identified 75% correctly. More than half of the males and females were identified remarkably well but some walkers were identified at no better than a chance level.

A statistical evaluation showed that height accounted for 70% of the variance and estimated center of mass even more. Spectral analysis indicated that walkers were more probably judged male when the walking sound contained more energy in the low frequencies with rapid spectral rising and falling. Female judgments were more likely when there was more energy in the higher frequencies with slow spectral rising and falling.

2.1.3 Perception of Object Properties by Sound

It has been long assumed that geometrical and material properties of objects could only be assessed by visual perception. However the physical parameters and spatial dimensions of object determine the sound that it generates when it is mechanically disturbed. This leads to the assumption that the acoustic structure is potentially informative about the object properties. There are many investigations showing that the perception of the object properties by sound alone is surprisingly reliable [12].

An early investigation on this psychoacoustic area was made by Warren and Verbrukke [54] in the mid-1980s. The impact sound of bottles and jars of different sizes were recorded as they fall on the ground and the listeners were asked to identify if the dropped object was bouncing from the ground or if it got broken. The success rate was 99%. The next step was to identify what in the sound structure allowed the identification. The result was that a bounce produces a single, damped, quasi-periodic pulse train and breaking produces an initial burst which is followed by multiple, damped, quasi-periodic pulse trains.

Carello [11] showed that listeners are surprisingly well able to estimate the length of a rod dropped on a floor. The experiments of Lakatos and his collagues [31] and Kunkler-Peck and Turvey [29] demonstrated that subjects can estimate the ratio dimension and the shape of either struck bars or plates. Listeners were also able to scale the hardness of a set of mallets striking cooking pans [20].

2.2 Survey of General Synthesis Techniques

2.2.1 Non-Physical Modeling

Until the 90s, sound synthesis was mainly based on non-physical methods. Frequency modulation (FM) [42] where frequency of oscillator is modulated with another oscillator is computationally efficient synthesis method. Sampling synthesis [42] is based on playing prerecorded sound samples and wavetable synthesis [42] is done by concatenating short samples of different waveforms.

Another non-physical method is granular synthesis [40]. It is based on the theory of Gabor [22, 23] which says that any possible sound can be formed by large number of elementary sonic grains. Granular synthesis is based on small samples of sound, which are combined in time-frequency domain to create any kind of sounds.

In quasi-synchronous granular synthesis [41], the synthetic sound is created by generating one or more streams of grains. The grains follow another with variable delays. In asynchronous granular synthesis [41] thousands of microstructures are combined to create macrostructures or clouds of sound. The clouds can be specified with various parameters such as start time and duration of cloud, grain duration, density of grains per second, amplitude envelope of the cloud, and possibly other statistical parameters.

A major problem in these modeling methods is that the physics-based control of synthesis parameters is very difficult. Quasi-synchronous granular synthesis could be a good model for relatively few clappers but in case of a large group, the computational load of the method may be too high. In asyncronous granular synthesis, clouds of hand claps could easily model a large group of clappers and the applause could also be synchronized. However the purpose of this work is to create efficient physically-based synthesis with control parameters that are directly connected to the sound producing event. Thus neither granular synthesis nor other non-physical methods can be considered as a eligible implementation techniques.

2.2.2 Physical Modeling

A problem of non-physical synthesis techniques is that tuning of synthesis parameters is difficult as it is not based on real-life physics. Physical models [51, 52] are being developed for musical instruments for two reasons: high-quality sound synthesis and research of acoustical properties. In the following the basic categories of physical modeling are presented.

Finite-difference methods can be used for solving partial difference equations that describe a vibration of a string, membrane or air in a tube. This method was first used in physical modeling by Hiller and Ruiz [24, 25].

Mass-spring networks by Cadoz [10] is a method where for example a vibrating string can be simulated as a mass-spring network by dividing a string into small segments that have a finite mass and elasticity.

In wave digital filters a mechanical mode is constructed for the part of the instrument or interest. Then the model is converted to an electrical circuit which is the discretized using the bilinear transform. This method has been used for example in the synthesis of wind instruments [50].

Digital waveguide modeling is based on a traveling wave decomposition of vibration in linear structures. It is a common method in the modeling of linear resonator parts of musical instruments, such as strings [53] and tubes [5].

In source-filter models a source signal, typically spectrally rich periodic waveform, is filtered with a time varying filter that can be controlled with parameters. Source-filter methods are usually classified as a non-physical modeling method but for example a voice production model from speech and singing synthesis is based on a physical modeling [44].

Modal synthesis is a technique in which the synthesis of this work is based on. It describes a linear system in terms of its modes of vibration. The synthesis is strongly related to vibration measurements of a specific structure or instrument. An early application of this technique was done by Adrien [1]. In his work, each vibrating mode has a resonance frequency, damping factor and physical shape specified on a discrete grid. Modal analysis was also used in the PhISEM model [14, 15] which is described in more detail in the following section.

In the case of this work, the problem of finite-difference, mass-spring, wave digital, and digital wavequide models is that a hand is a complicated structure and physical modeling of it would be too hard. Therefore the problem is assessed through modal synthesis as the resonance modes of hand clapping can be easily measured.

2.2.3 Physically Based Models and "Cartoonification"

Physically-based low-level models [4, 43] are based on a simplistic physical mechanism involved in the sound generation process. They include the description of the resonating structure as well as simple interaction modalities. They offer a functional way to synthesize naturally behaving sounds from computational structures that can easily interact with the environment and respond to physical input parameters. Although real sounds serve as an orientation, the realistic imitation is not the goal of synthesis. Some simplifications which preserve or even exaggerate some acoustic properties, while losing others that are not so important are often used. These "cartoonifications" are often perceptually sharper than the realistic simulations and they also increase the computational efficiency.

The sound synthesis and computer graphic communities have increasingly used physical models when the goal is a natural dynamic behavior. The secondary effect is that the models are easy to control as the physical parameters can be tuned. Futhermore as the parameters are based on the physics of the model, they can be interactively controlled by sensors and actuators. The physics based sound model can be also done in several degrees of fidelity. Synthesis models that are significantly simplified can be used for example in portable devices such as mobile phones or electronic games. Models with higher details are used if more fidelity is needed and more computational power is available.

2.3 Importance of Control in Physics-Based Sound Synthesis

While the synthesis techniques have achieved good results in reproducing musical and everyday sounds, most of the techniques focus only on perfect synthesis of isolated sounds. Thus the fact that most of the expressive content of sound messages comes from the appropriate articulation of sound sequences is often ignored. Sound control [43] is easier if the sounds are generated with physics-based synthesis techniques that give access to the control parameters which are directly connected to the sound source characteristics.

A good example of the interplay between physics-based synthesis and physics-based control is a wind chime synthesis introduced by Lukkari and Välimäki [33]. Modal synthesis is used to model the tubes of a wind chime. Then a physics based stocastic process is used to trigger sound events caused by the clapper hitting tubes. The energy of the system is exponentially decaying and it can be augmented with a wind force parameter. The propablity of a hit occuring is then counted from the energy of the system.

2.3.1 The Sounding Object Project

One part of the Sounding Object project was focused on investigating the control models for sound events [7]. Dahl [16] investigated the relations between music performance and body motion. Musicians use their body in many ways to produce sound. The research of Dahl was focused on movements and timing of percussionists when playing an interleaved accent in drumming. It showed that drummers prepare for accented stroke by raising the hand higher. This leads to more velocity at the striking point. Another study showed that drummers prefer auditory feedback more than tactile [17]. These observations on percussionists' behaviour can be used to model the control of any kind of impact sound model where human action is used to manipulate the sound source, of course including hand clapping.

Another research on control has demonstrated that there are several performance rules that allow the expression of emotions in computer-controlled music performances [8]. Rendering of emotions in speech and music can be achieved by controlling only a few of these rules. This cartoonification of control can be used together with cartoonificated sound models to obtain simlified but realistic synthesis of sound event.

An implementation of control model for crushing, walking and running sounds was also presented [18, 19]. The implementation was based on bottom-up design strategy. It started from a existing physically-based impact model. Then this impact model was connencted to dynamic and temporal stocastic charasteristics of chrushing events. To trigger these crushing events, control rules realizing typical walking and running patterns were used.

The control model of walking and running was based on three musical performance rules:

1. *Legato* rule means that there is a overlap time when two notes are played simultaneously. This correspond to the overlap time in walking when both feet are on the ground [9].

2. *Staccato* rule means that there is a silent time interval between two notes. In running this is equivalent to the time interval when both feet are in the air simultaneously [9].

3. *Final Ritard* is a rule that is normally used in performances of Baroque music when the tempo becomes slower *(ritardandi)* before the end. It can also be used when simulating a stopping runner [21].

2.3.2 PhISEM

The PhISEM (Physically Informed Stochastic Event Modeling) algorithm introduced by Cook [14, 15] is based on pseudo-random overlapping and adding of small grains of sound. Complex three-dimensional multiple particle models, such as maracas, wind chimes, ice cubes in a glass, etc, can be modeled using the idea of particles contained in a shell.

The implementation of these multiparticle models uses rapidly decaying, exponentially enveloped noise for individual collision events of the particles inside the system. One or more two-pole filters are used to model the resonance of the sound event. As the system is linear, i.e. sum of exponentially decaying random noises is equal to single noise source multiplied by a decaying value, only one exponential decay function and one noise sound source is needed. The sound envelope is increased additively by each new collision, as indicated by the Poisson process calculation. An exponential decay models the net system energy, and determines how much energy is added with each collision.

The PhISEM synthesis algorithm is quite simple [14, 15]. It only requires two random number calculations, two exponential decays, and one set of resonant filter calculations to produce one sample of synthetic sound. It is also flexible for control of the number of particles and system resonances. Cook also introduced the parameters with which the synthesis can be tuned to model many percussive instruments like sekere, cabasa, guiro, tambourine, and sleighbells or even nonmusical sounds like water drops and bamboo wind chimes.

2.3.3 Ensemble Control Modeling

In order to model the audience giving applause, the control is in a important role. The physics of rhythmic applause has been investigated by Néda and his collagues [36, 37]. Their research was focused on a phenomenon where the applause of audience changes from thunderous unsynchronized clapping to synchronized rhythmic applause.

The conclusion of their study was that the spectators had two very distinct clapping modes: the normal high frequency clapping mode and the low frequency clapping mode which is used in synchronized clapping. The clapping interval is roughly double and the dispersion of the natural frequencies of the spectators is half in synchronous clapping. The adaptation between the modes can be explained by the Kuramoto model of coupled nonlinear oscillators [30].

The main result of the Kuramoto model is that for a population of globally coupled non-identical oscillators a partial synchronization of the phases is possible whenever the interaction among oscillators exceeds a critical value. The critical coupling for synchronization is directly proportional with the dispersion of the frequency distributions of the oscillators. When the dispersion is decreased the value of critical coupling decreases too. When the audience is clapping with high frequencies the frequency distribution is wide and the value of critical coupling is large [36, 37]. The coupling between the spectators is lower than the critical value and there is no synchronization.

The audience subconsciously aims synchronized applause by doubling the clapping period [36, 37]. This reduces the dispersion of natural frequencies and also the value of critical coupling decreases allowing the synchronization. However, when the synchronization is achieved the total signal level becomes lower. Enthusiastic spectators want to increase the total signal level by increasing the clapping frequency. This increases the dispersion of natural frequencies and the value of critical coupling leading to the lost of synchronization.

Chapter 3

Physics-Based Synthesis of One Clap

3.1 Measurements

In order to make a physics-based synthesis, the sound producing process must be analyzed. There were some shortcomings in the measurement procedure of Repp's study. In order to get more reliable results some measurements were made.

The measurements were made in an anechoic chamber. Two AKG 480B microphones were positioned at the distance of one meter from the subjects' hands. The first microphone which was connected to the right channel of Yamaha 01V mixer was positioned directly at the front of the subject and the second microphone which was connected to the left channel was at the angle of 60 degrees on the left side of the clapper. The mixer was connected to a HP Omnibook XE[3] laptop computer with a Digigram VX Pocket V2 soundcard. 44100 Hz sampling rate was used in measurements. Care was taken to set the recording level so that no peak distortion occurred.

The movement of the body when clapping hands caused some vibration on the mesh of the anechoic chamber. We tried to reduce this vibration by placing the subject at the mounted ground. However, some vibration remained and it had to be removed by filtering the recordings with a fifth order Chebyshev Type I highpass filter with the cut off frequency of 100 Hz.

The test was made with three subjects: the author and two colleagues. In order to get more accurate information about the spectral shape of different clapping modes, a sequence of five claps in each clapping mode was recorded. Same clapping modes were used as in Repp's study [39]. The positioning of hands was photographed for each clapping mode. These photos can be seen in Figure 2.1.

For later research on statistical control modeling of one or several clappers with different levels of enthusiasm some extra measurements were made. First the subject was asked to do a sequence of natural hand clapping without any instructions. Then a sequence of very enthusiastic and very bored clapping was recorded. After the different modes of clapping were recorded the subjects were asked to repeat a sequence of normal, enthusiastic and bored clapping but now in clapping mode A2.

3.2 Analysis of Test Results

3.2.1 Spectral Analysis

The average spectra of five claps in all clapping modes can be seen in Figure 3.1. The spectra were counted using Discrete Fourier Transform. Comparing these results to Repp's research we can observe the similarities in spectra. As Repp pointed out, the modes A1, A1+ and A1- seem to have loadings on low-frequency peak factor while modes A3 and P3 have loadings on mid-frequency peak factor. The average spectra of any individual clapper did not differ significantly from others.

3.2.2 Extracting Resonance Peaks

As the synthesis will be implemented using resonators, the characteristics of frequency peaks must be extracted. The design of these two pole filters is based on the center frequency f_c and -3 dB bandwidth B.

Proper tool for representation of a spectum is linear predictive coding (LPC) which has been used in speech coding applications since late 60's [3]. It is an all-pole filter which models the peaks of the spectrum. The problem of linear prediction in wideband applications is that it uses too much coefficients to model high frequencies which are not so important in terms of human hearing. This can be overcome using frequency-warped linear predictive coding or WLPC [26]. In warped signal processing techniques, the spectral representation is modified by replasing the unit delay elements with first-order allpass filters. The spectral resolution of WLPC is much closer to the frequency resolution of human hearing. Another way to improve LPC results is to use linear prediction with low-frequency emphasis, or LPLE, which was introduced by Alku and Bäckström [2].

However, in analysis there is no harm in using conventional LPC. The selection of LPC order is the difficult part. If the order is too high, the peaks start to split to several peaks. If the LPC order is too low, the result is not accurate enought. A good method for founding proper linear prediction order is to count the frequencies corresponding to pole angles for every LPC order and examine when the poles start to split [28]. Figure 3.2 illustrates this method. The magnitude spectrum of one hand clap is plotted above for comparison. The map below shows how the poles split to new pole pairs as the linear prediction order

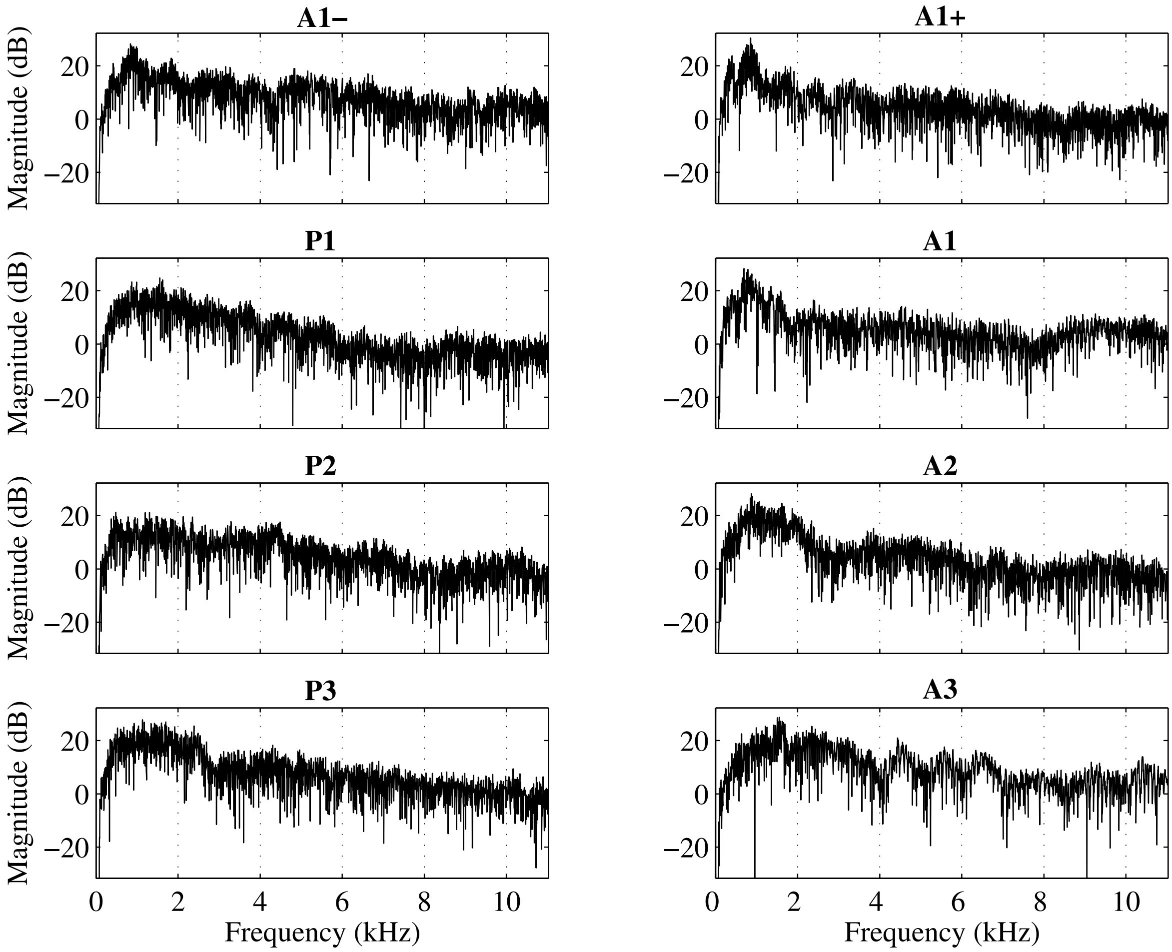

Figure 3.1: The average of five recorded clap spectra of all clapping modes.

increases. Splitting can be observed also by counting the pole radius of the pole that is nearest to the frequency of interest. When a pole splits, the pole radius starts to fluctuate.

A good model order for our purposes is roughly something between 50 and 100. Also higher orders can be used even though then the LPC starts to model the spectrum with high detail rather than just the envelope. The strongest resonance peaks were isolated using 81th order LPC and the highest peaks and their bandwidth were detected for each clap. An example of a resonator fitted to LP spectrum of a single hand clap in mode A3 can be seen in Figure 3.3. Also FFT of the signal is plotted with small offset for comparison.

In Figure 3.4 we can see the mean value and standard deviation of center frequency f_c, bandwidth B and gain g of the strongest resonance for all clapping modes. Table of numerical values of these statistics can be seen in Appendix A.1. The statistics are counted from each clap of each subject (i.e. 15 claps in total). As the number of claps is so small, it

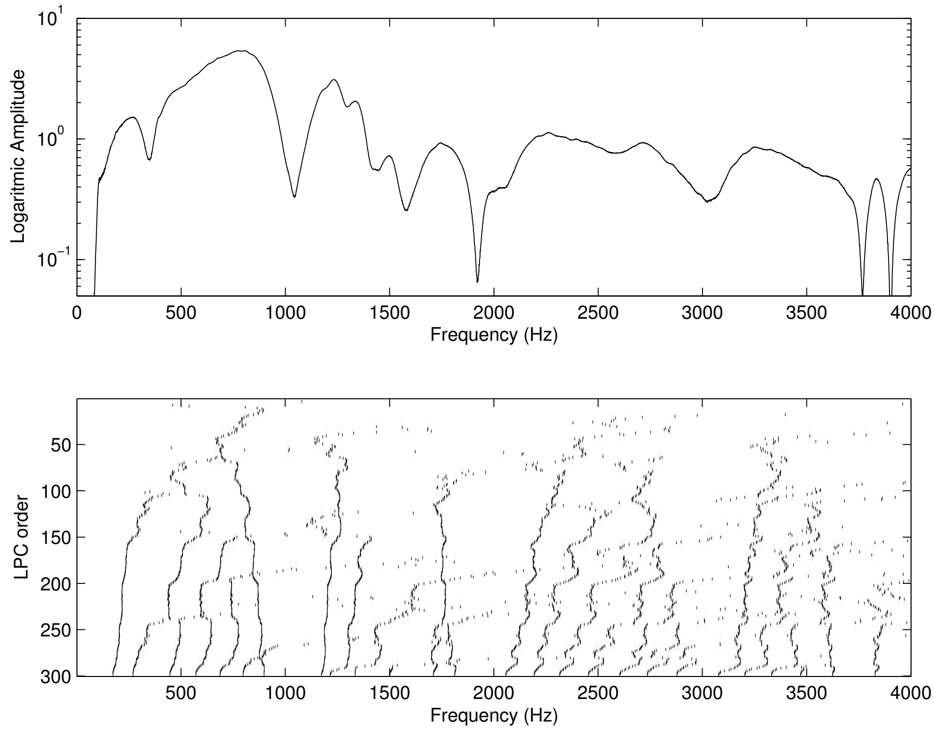

Figure 3.2: Magnitude spectrum of a single clap and the frequencies corresponding to pole angles obtained from linear prediction of varying order.

is difficult to analyze the shape of the distribution. The Gaussian distribution was assumed to get indicative results for distribution. Again it can be observed that modes A1, A1+ and A1- have their strongest resonances below 1 kHz and modes P3 and A3 at 1.5 kHz.

These statistics can be used to model the resonances that occur when clapping hands. As Repp's study showed, the subjects were able to recognize their own hand clapping relatively well. Based on this assumption the future resynthesis will be based on modal characteristics derived only from the author's hand clapping recordings. The statistics of the strongest resonance in all author's clapping modes can also be seen in Appendix A.2. There can be observed that the variation between different clapping modes is even more obvious.

3.2.3 Inverse Filtering

Once the main resonances of hand clapping recordings are extracted, the main resonances can be filtered out of the original signal. This spectral flattening technique called inverse filtering [46] is widely used in source-filter methods like speech synthesis and coding.

The resonator filter is first inverted i.e. the numerator and denominator are transposed. After that the filter is applied to the original signal from which it was extracted. This results

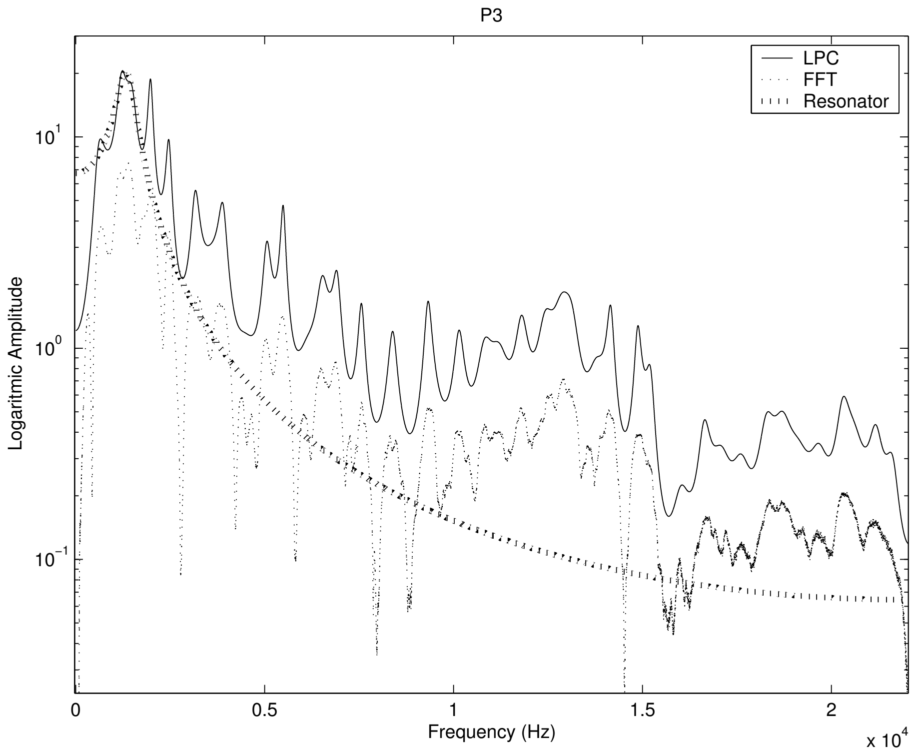

Figure 3.3: LP spectrum of a single clap, a resonator fitted to model it and for comparison the FFT spectrum of the clap.

a source-signal whose spectral shape is expected to be similar in every clapping mode as the resonance peak characteristic to each mode is filtered out. Then an average spectrum of every inverse filtered clap is counted and shape of spectrum can be observed. Figure 3.5 shows an average spectrum of inverse filtered signals and a second order IIR filter fitted to model it. This bandpass filter can be used in final synthesis.

3.2.4 Time-Domain Analysis of a Single Clap

In order to choose a good excitation signal for resonators, also some time-domain characteristics will be derived. The attack time is the time from the start of the pulse to its peak value and decay time is the time from peak value to the end of the pulse. Envelope detection can be used to derive these values more accurately but indicative results are easily obtained by finding the highest sample value of the signal and comparing it to the start and end time of a clap. In Table 3.1 the average attack time and length in samples of each clapping mode is showed. As can be expected the length of clap is longer in modes with sharper resonances at low frequencies.

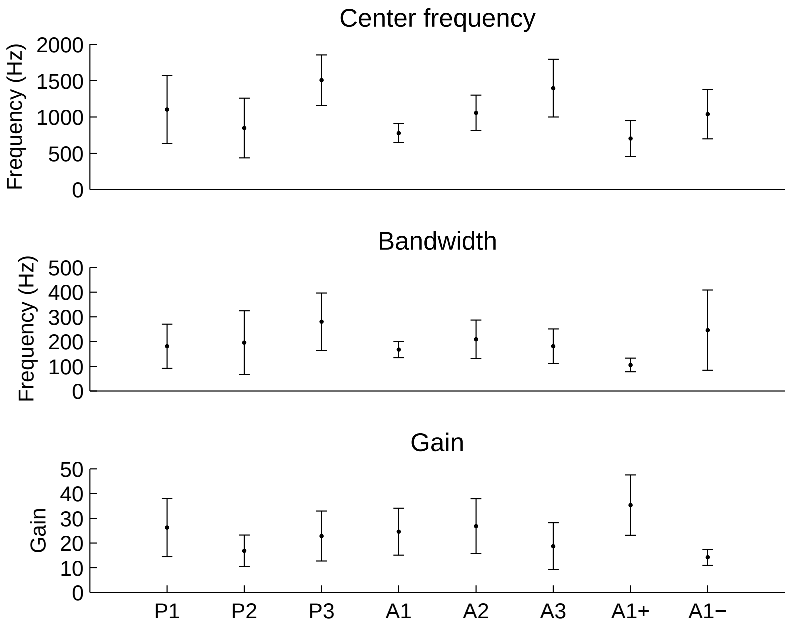

Figure 3.4: The mean value and deviation of center frequency, bandwidth, and gain.

Table 3.1: The average attack and decay time and clap duration of each clapping mode in samples (f_s=44100).

	Attack time (in samples)	Attack time (ms)	Decay time (in samples)	Decay time (ms)	Clap duration (in samples)	Clap duration (ms)
P1	208	4.7	497	11.3	687	15.6
P2	132	3.0	382	8.7	514	11.7
P3	154	3.5	216	4.9	370	8.4
A1	176	4.0	418	9.5	594	13.5
A2	170	3.9	266	6.0	436	9.9
A3	140	3.2	229	5.2	369	8.4
A1+	220	5.0	497	11.3	717	16.3
A1-	59	1.3	321	7.3	380	8.6

3.3 Cartoonificated Resynthesis

At this point it is good to remind that the goal of this work is not the exact resynthesis of recorded claps. In fact the clapping of hand in an anechoic chamber does not even sound as a clapping of hands that we are used to hear – it is only a very short impulse. The sound that we are used to hear consists mostly from reverberations in a room so it should actually be considered as a room impulse response rather than a clap. Keeping this in mind the goal

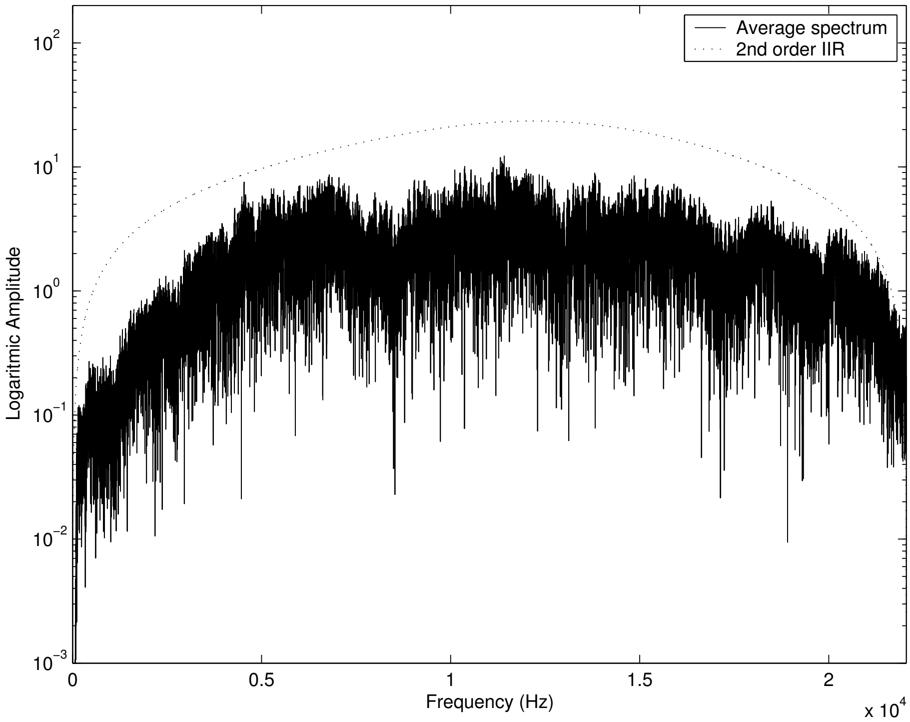

Figure 3.5: Average spectrum of every recorded clap (8 modes, 5 claps in each) and a second order bandpass filter (with small offset) fitted to model it.

is a cartoonificated synthesis of hand clapping.

3.3.1 Resonators

A simplified model of percussive everyday sounds like hand clapping can be easily done using white noise bursts and resonators. The resonators are two-pole filters where the resonance is caused by a pole near the unit circle. Designing a resonator filter can be done by following steps [47]:

The bandwidth B and the resonant frequency ψ were derived from spectral analysis of test results. Then we calculate the pole radius R from the bandwidth.

$$R \approx 1 - B/2 \tag{3.1}$$

After that we calculate the cosine of the pole angle θ:

$$\cos\theta = \frac{2R}{1+R^2}\cos\psi \tag{3.2}$$

Then the gain factor A_0 is calculated. It makes the magnitude response unity at resonant frequency ψ.

$$A_0 = (1 - R^2)\sin\theta \tag{3.3}$$

Finally using the filter equation we get:

$$y(n) = A_0 x(n) + 2R\cos(\theta)y(n-1) - R^2 y(n-2) \tag{3.4}$$

In order to model multiple resonances, multiple parallel resonators can be used. Their gains are adjusted by multiplying filter coefficient A_0 with a gain g which was also derived from analysis.

3.3.2 Choosing an Excitation Signal

Several types of excitation signals can be considered. The easiest and computationally most efficient excitation is exponentially decaying white noise, which was also used in PhISEM-model introduced by Cook [14, 15]. However, if exponentially decaying noise is used on isolated claps the result is not perceptually very good, because there is no attack phase in the resulting signal.

Better results are obtained if only exponentially rising noise attac excitation is used. Resulting signal contains an attac part which is the noise burst that comes when two hands hit together. After this comes the decay phase which matches to the resonance of the resonator (i.e. the cavity formed between hands). Only problem in this model is that the resulting sound is very short as it models only a clap event in the anechoic conditions.

A synthetic clap propably sounds better if it is made a little longer. If the exponentially rising noise of the attack phase is combined with an exponentially decreasing noise of the decay phase the resulting clap is perceptually better even though the sound is slightly metallic. However, this applies only for a single clap. In the case of a clapping sequence, the model in which only exponentially rising noise is used as a excitation is clearly better.

The excitation noise is then filtered with the bandpass filter obtained from inverse filtering. This results a good excitation signal for every clapping mode.

3.3.3 Synthesis results

The resynthesis of one clap was implemented in Pure Data (Pd) [13, 38]. It is a real-time graphical programming environment for audio, video, and graphical processing. Simplified block diagram for synthesis process can be seen in Figure 3.6 and a Pd patch in Figure 3.7.

When the system is triggered, new resonator coefficients are calculated based on mean value and standard deviation obtained from analysis. Clapping mode can be changed from the message boxes at the down of a patch. New coefficients are updated for each clap so that there is some variation between each clap. Trigger also launches the envelope generator

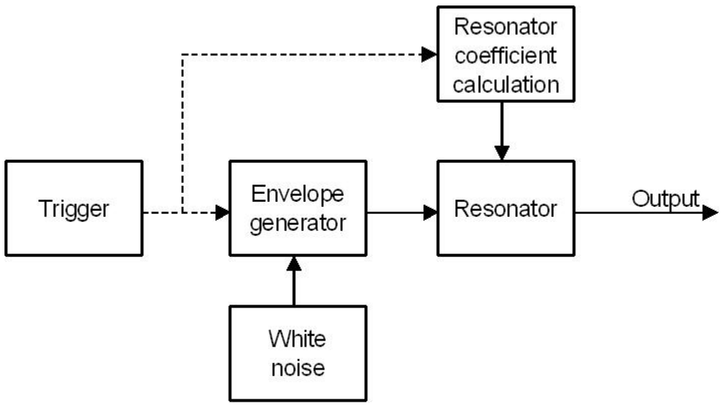

Figure 3.6: Block diagram of synthesis of single hand clap.

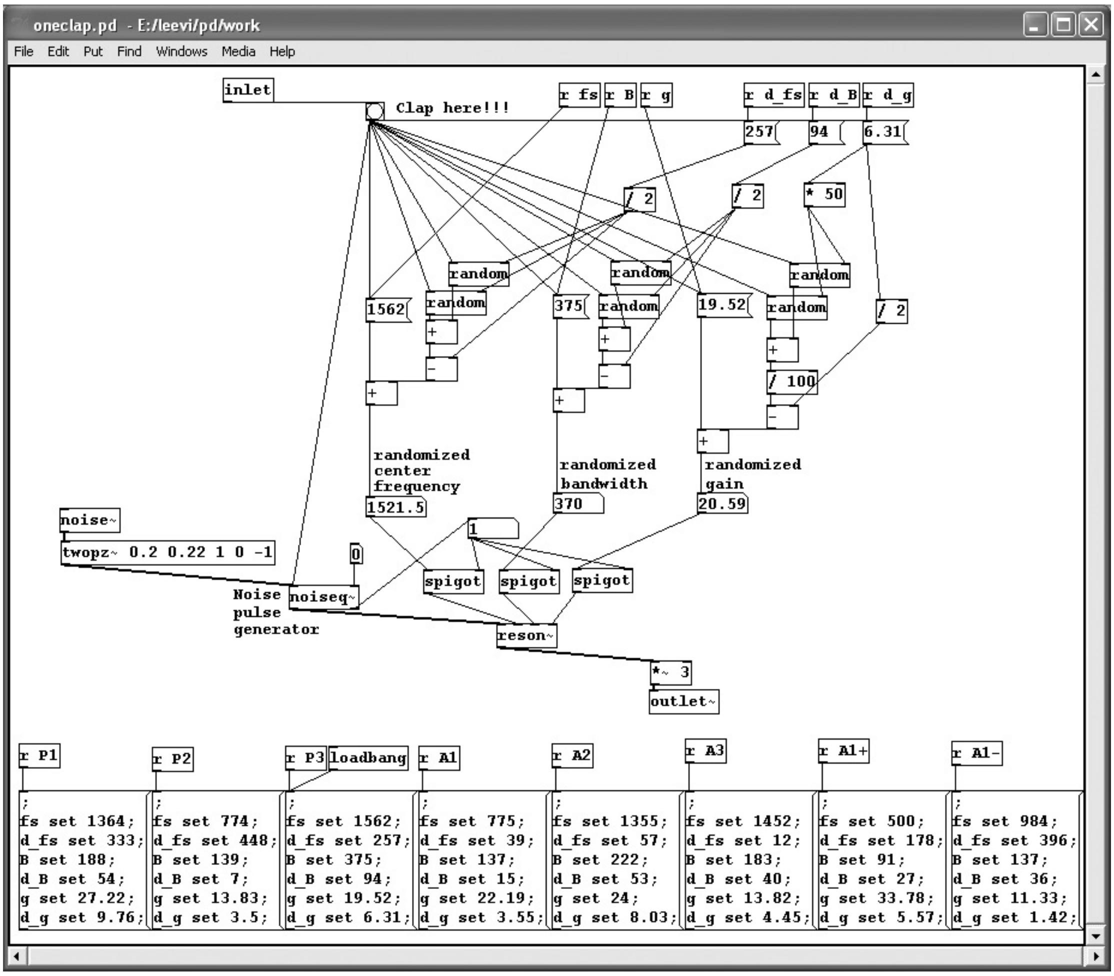

Figure 3.7: Pure data patch for a synthesis process of one clapper.

which passes the enveloped noise signal to the resonator. Both, the envelope generator and the resonator are Pd externals implemented in C programming language. The source code can be examined in Appendices B.1.2 and B.1.1.

Figure 3.9 shows two synthetic claps that were generated with excitation signal that has both the attack and the decay phase. Figure 3.10 shows two synthetic claps with excitation signal that has only attack phase. For comparison also two original claps are plotted in Figure 3.8. Even though all claps are in mode A3, some variation in spectral shapes as well as in time domain signals can be observed.

The spectral shape of synthetic clap is very similar to the originals. In general the main resonances of synthetic claps are slightly sharper. The resonance is even sharper in synthesis model where there is only attack phase in the excitation signal. This sharpness is caused by the cartoonification in the synthesis model: it was assumed that the sound of hand clapping consists only from one resonance and only the strongest resonance was extracted. Also the order of LPC might have been too high or too low. However, sharper resonace makes the sound more recognisable, and that is what is important. Spectrum of synthetic claps also contains lots of notches. These are propably caused by the shortness of excitation signal. The excitation is only 4-10 ms (i.e. 180-440 samples) long white noise pulse. The shortness may result that the spectrum of the white noise is not absolutely flat.

In time domain the difference is more obivious. Synthetic claps are remarkably longer than the original ones. The length of the attac phase (and possible decay phase) in excitation signal is based on the characteristics of the original claps. The duration of the synthetic clap is then prolonged by the resonator.

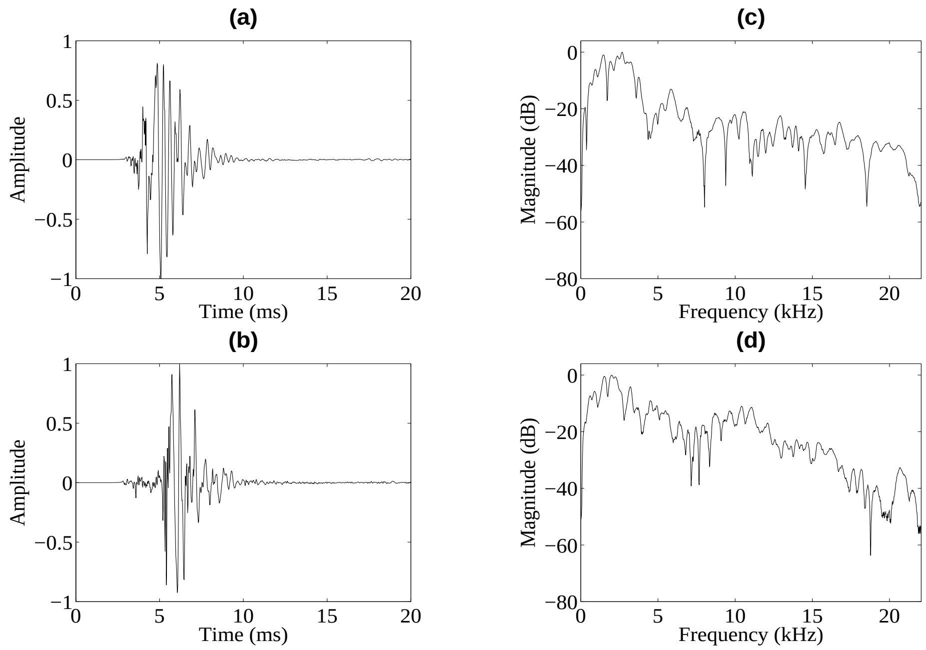

Figure 3.8: Time (a and b) and frequency (c and d) domain representation of two recorded claps.

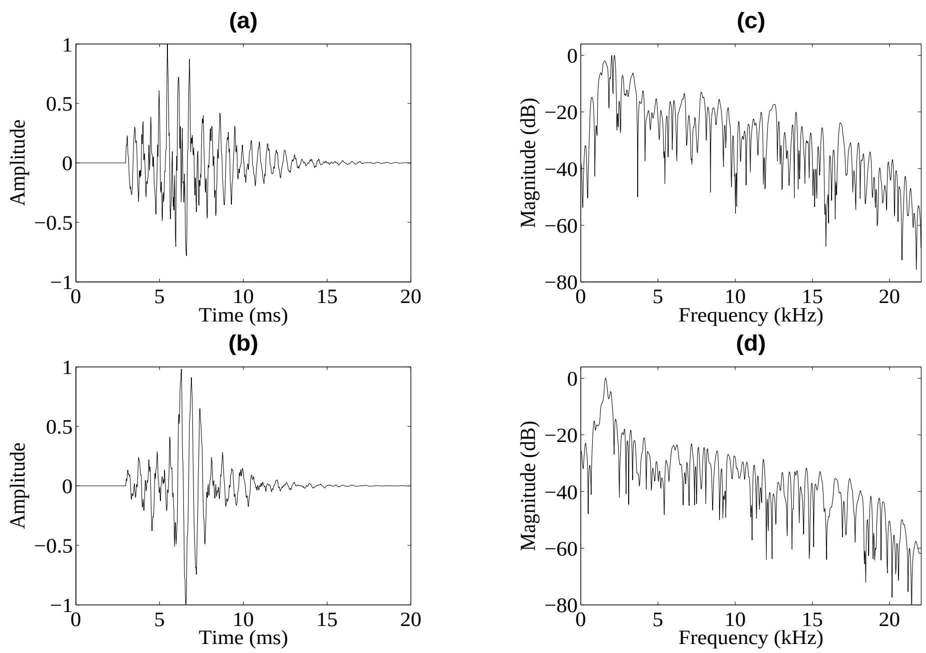

Figure 3.9: Time (a and b) and frequency (c and d) domain representation of two synthetic claps with excitation signal that has attack and decay phase.

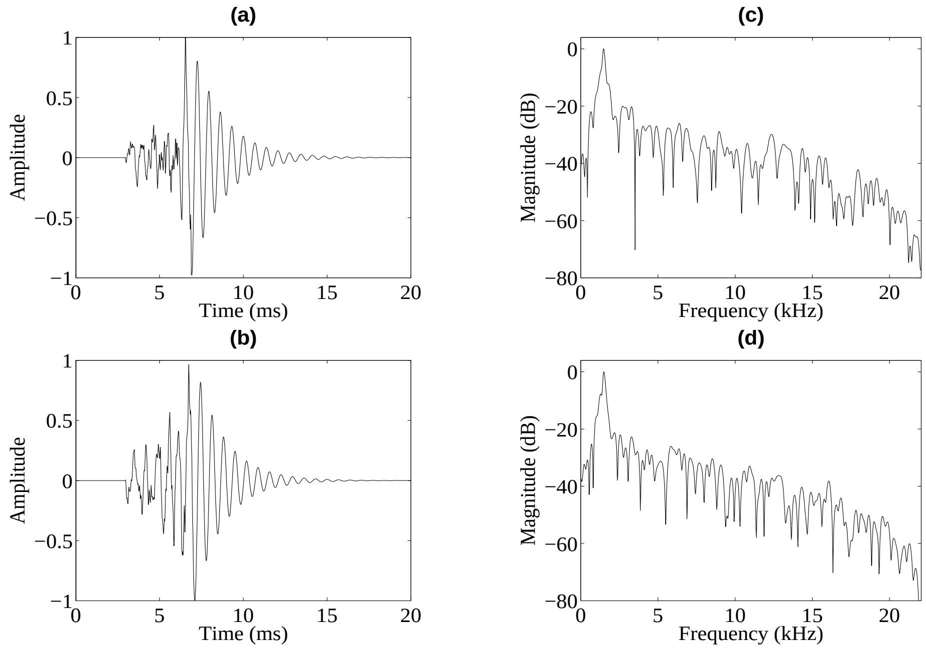

Figure 3.10: Time (a and b) and frequency (c and d) domain representation of two synthetic claps with excitation signal that has only attack phase.

Chapter 4

Statistical Control Modeling of One Clapper

Now that we have a synthesis model for a single clap event the next task is to create a realistic control for these events. As was said in Chapter 2, a good control model for sound events is at least as important as the synthesis model for an isolated sound.

This chapter focuses on creating a control model for a single clapper. Of course several clappers can be modeled using the same model, but it will lead to memory consuming model when the number of clappers rises too high. At the same time, if the number of people is high, it is difficult to distinguish the fluctuation of the clapping rate of an individual clapper. Thus, it is not very smart nor useful to use many of these models in parallel. Naturally, this model could also be used offline to model the sound of various clappers when deriving more efficient model for higher amount of people. However, the goal of this part of the research is to create a credible imitation of a single person clapping hands.

4.1 Analysis of the Statistics of a Clapping Sequence

4.1.1 Clapping Rate

When making measurements to model different clapping modes, also some measurements for statistical modeling were made. First the subjects were asked to clap their hands naturally, as they would after an average performance. Then they were asked to make a very enthusiastic and after that very bored sequence of hand clapping. After measuring the clapping modes, subjects were asked to make these three clapping sequences again, but now using clapping mode A2. An example of a hand clapping sequence can be seen in Figure 4.1 Results obtained from our measurements can be seen in Table 4.1.

These results give some direction for the clapping rates with different levels of enthusi-

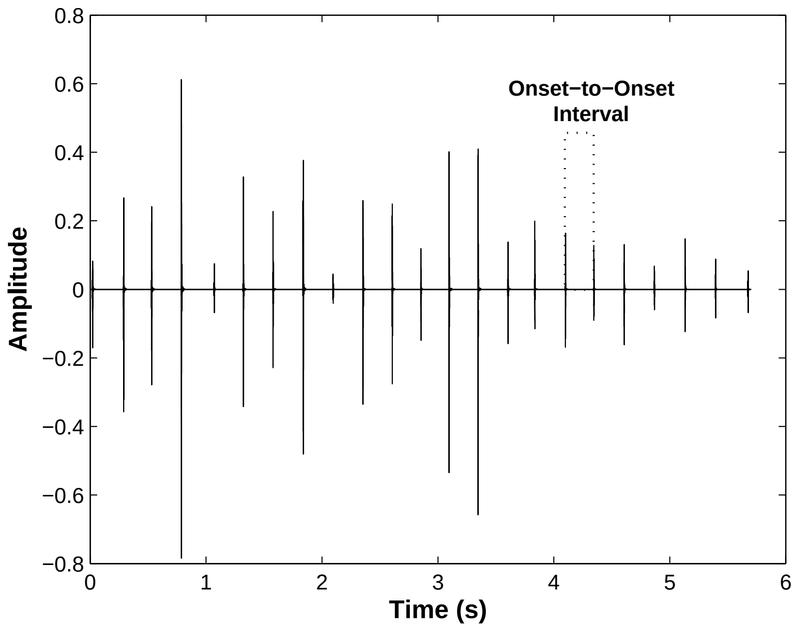

Figure 4.1: Example of hand clapping sequence

Table 4.1: Onset-to-Onset Intervals obtained from measurements

	Natural	Enthusiastic	Bored
Average OOI	403 ms	323 ms	612 ms
Minimum OOI	316 ms	232 ms	362 ms
Maximum OOI	610 ms	547 ms	829 ms
Standard deviation of OOI within a sequence	25 ms	21 ms	29 ms

asm. Anyhow, as the number of subjects was only three and only six sequences of clapping was measured for each clapper these results can not be considered very reliable. Thus we should also pay attention to clapping rate measurements by Repp [39] and Néda et al. [36, 37].

In Reppt's research the number of subjects was 20. Average onset-to-onset interval (OOI) in his measurements was 250 ms ranging between 196 ms and 366 ms. Standard deviation varied between 2.8 ms and 13.6 ms (1% and 5%) with average of 6.8 ms (2.7%). There was a small difference on clapping rates between genders. Males clapped slightly slower (average OOI = 265 ms) than females (average OOI = 236 ms). Also articles by Néda et al. [36, 37] give some interesting information on clapping rates. They measured the clapping rate of 73 subjects. First the subjects were asked to clap naturally as they would after a good performance. Then they were asked to clap in the manner they would do during the synchronized applause. Average Onset-to-Onset Interval of natural clapping (OOI_{NAT}) was roughly 250 ms and in synchronized mode (OOI_{SYNC}) about 500 ms.

Based on all of these measurements some rough statistics for the implementation can be derived. The clapping rate was chosen to vary between 4.17 Hz for enthusiastic clapping (OOI = 240 ms) and 2.5 Hz for bored clapping (OOI = 400 ms). Even though the measurement data shows that the OOI for bored clapping averages to the 600 ms it sounds more realistic if the clapping rate is a little bit faster. Especially in the case of bored clapping, our subjects tend to exaggerate the slow clapping rate.

4.1.2 Performance Rules

In addition, some other characteristics that are typical for humans can be found when analyzing the recorded clapping sequences. Variation of the OOI is not constant during the whole clapping sequence. For example, at the start of a sequence it takes some time for subjects to find their convenient clapping frequency. That is why the variation of the OOI is usually larger at the start of a sequence. Also if the subjects clapping rhythm is disturbed for some reason the variation of OOI may increase for a while and then decrease again. This interruption may be caused for example by fatigue or attempt to synchronize the clapping rate to other spectators. An example of a clapping sequence where the variation of OOI is larger at the start and end of a sequence can be seen in Figure 4.2. Length of a bar in figure stands for a time interval between claps (OOI). In this sequence subject was asked to clap naturally. Clapping rhythm is also disturbed just before the end of a sequence.

The variation of the clapping rate is not just a randomized variation. It can be seen that the clapping rate fluctuates so that it is sometimes faster and sometimes slower. In musical performance *accelerandi* (Engl. accelerating) and *rallentandi* (Engl. slowing down) are used to model these changes of velocity [9]. They can also be used in control model of a clapping sequence to model the fluctuation of clapping rate.

Especially at the end of a clapping sequence the tempo seems to slow down a little. This indicates to the *Final Ritard* performance rule that was used in control model of walking and running sounds to model the decreasing tempo that precedes the stopping [19]. This phenomenon can be seen very well in long enthusiastic clapping sequences where the subject has problems on maintaining the fast clapping rate. This decay of clapping rate caused by becoming exhausted can also be considered as one reason for the transformation to synchronized clapping. Néda et al. [36] only proposed that the reason for the transformation is because the audience wants to clap synchronizedly and therefore double their clapping rate. An example of fluctuating clapping sequence can be seen in Figure 4.2.

These phenomena should be implemented and even exaggerated in the control model of a single clapper.

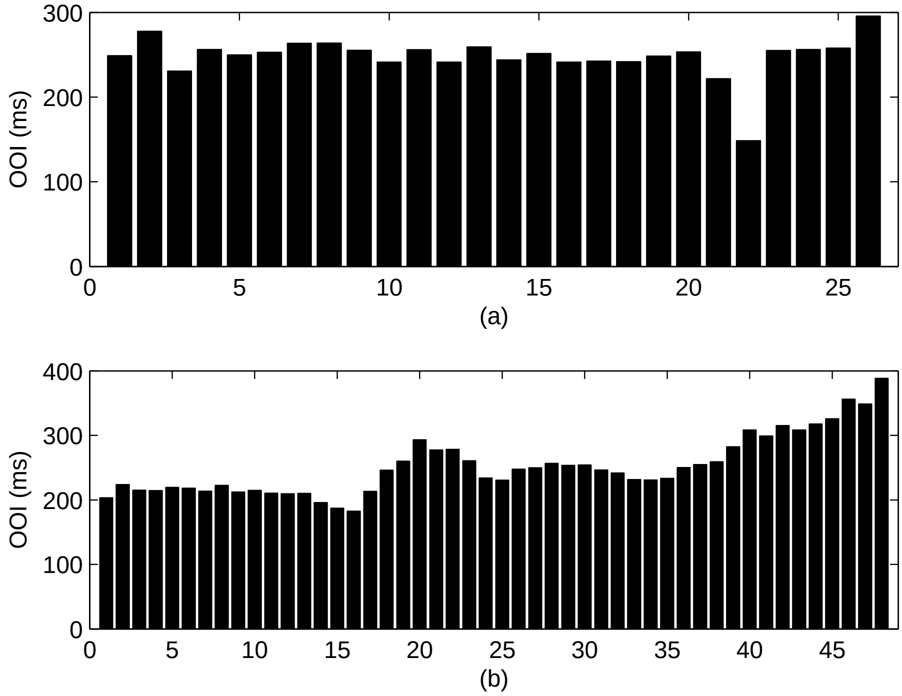

Figure 4.2: (a) Example of the OOIs of a clapping sequence where the variation of OOI is larger at the start. (b) Example of the OOIs of a clapping sequence with decreasing tempo at the end (*Final Ritard*).

4.2 Implementation

4.2.1 Control Model for a Single Clapper

The control model for single clapper was again implemented in Pd [13, 38]. User can control the length of a clapping sequence as well as the level of enthusiasm. Clapping rate varies between OOI = 240 ms for enthusiastic clapping and OOI = 400 ms for bored clapping. Variation of OOI is slightly exaggerated to 10% of the clapping rate. The variation is assumed to have a triangular distribution. Pd patch called *oneclapper* for a control model of a single clapper can be seen in Figure 4.3.

Implementation of the variation of the clapping rate is based on Pd's *pipe* object. It is a delay line whose length can be changed. When the system is triggered, a bang message is send to the delay line. When it comes out it triggers the synthesis model presented in previous chapter. After that a new length for delay line is calculated and the bang message is feed back to the delay line.

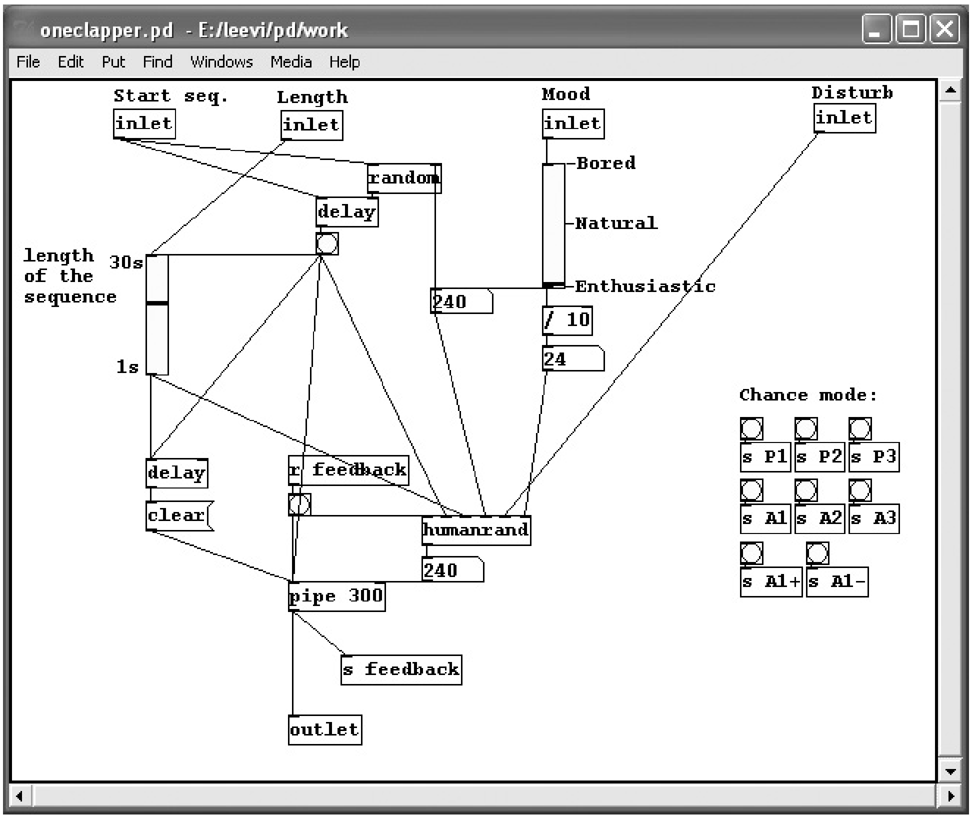

Figure 4.3: Pure data patch of a control model for one clapper.

To model the variation of a clapping rate another Pd patch called *humanrand* was made. It calculates the length of a delay line that is used in *oneclapper* and it can be seen in Figure 4.4. The measurements showed that the variation is larger at the start of a sequence. To model this phenomenon the variation of OOI is doubled during the first two second of a clapping sequence. This rule is implemented at the left side module of *humanrand*. The clapping rhythm can also be disturbed during the clapping sequence. When the clapper is disturbed, the variation is quintupled for one second. The disturbing is implemented at the up right corner of a *humanrand*. Below that is a part that controls the variation of a clapping rate.

Other phenomenon that is typical for a human clapper is the slowing of clapping tempo before the end of a sequence *(Final Ritard)*. This was modeled so that during the last third of the sequence, OOI is increased after each clap by 2% of the original clapping rate. Thus, the average time interval between claps is increasing linearly. *Final Ritard* is implemented at the center of the *humanrand* patch.

It is important that the control model of a clapping sequence is implemented separately

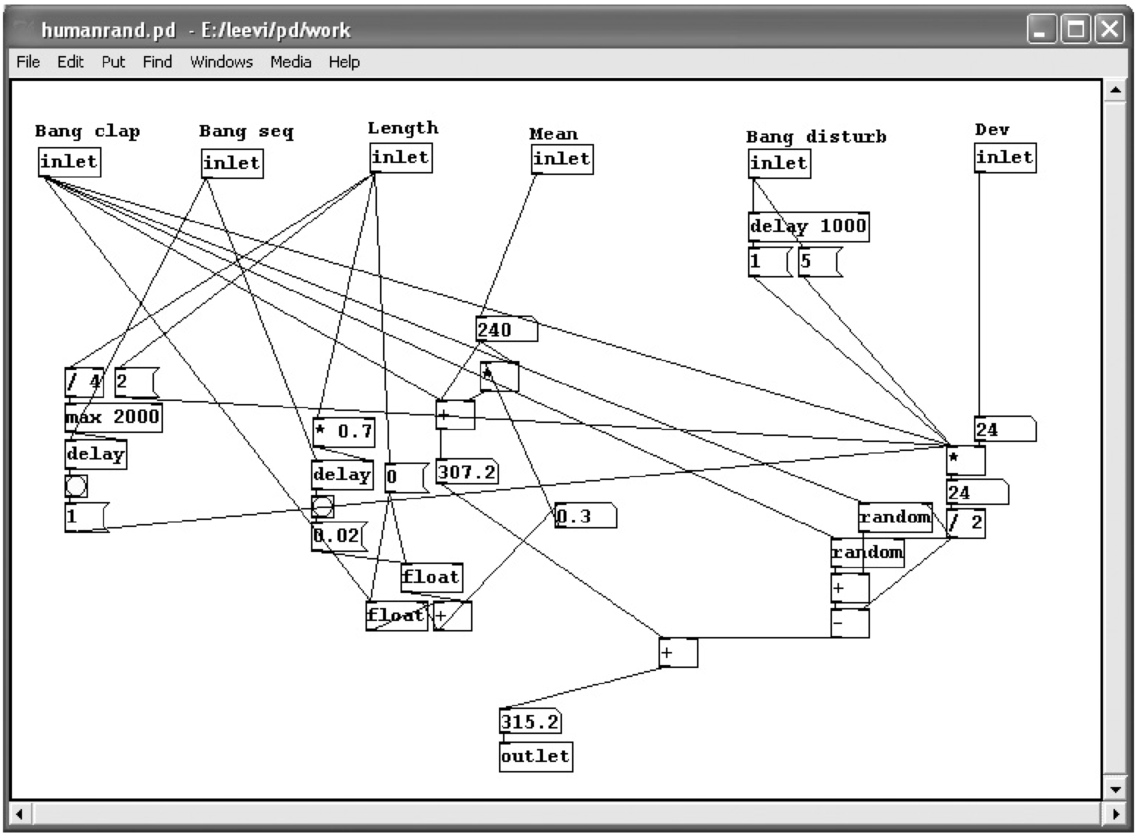

Figure 4.4: Pure data patch that is used to create a humanlike variation of clapping rate.

from the synthesis model of a single clap. If they would have been implemented together, also the control information should be calculated for each audio sample which means unnecessary increase of the computational load.

4.2.2 Improvements: Spatialization and Reverb

At this point we have a synthesis model for a single clap and a control model for a single clapper. The results are fairly credible even though they model the clapping in anechoic conditions. Naturally it can be expected that if the synthesis results are combined with a good spatialization, the results are notably improved. If the number of clappers is increased, the importance of room acoustics increases. In this work the main goal is not to create realistic room simulation and the topic is only shortly examined. For the readers who are interested in that line of research, Begault has written a good book to start with [6].

An important thing that has a significant effect in the case of several clappers is the spatial position of clappers and listener. If room acoustics are not considered the result will model several people clapping in same position of anechoic chamber. It is clear that this will not sound realistic. The first improvement is to give each clapper a position in this anechoic space so that the sound level is lower for the clappers that are farther away. Now each clapper has its own delay path and attenuation which corresponds to the distance to the listener.

The most important part for the spatial impression are the early reflections which reach the listener laterally [56]. These reflections give an impression of the distance to the sound source as well as shape and size of the room. For our needs it is sufficient to calculate only the first order early reflections (i.e. those that come directly from each wall). Early reflections can be easily modeled using virtual image source method. It is based on forming image rooms with secondary image sources. Then all sources are summed with corresponding delays and attenuations.

This simple spatialization makes the synthesis sound a bit better. The problem is that each clapper needs its own tapped delay line. This leads to very memory consuming model if the number of clappers rises too high.

Other option is to use reverberation algorithm. *Freeverb˜* [34] is a Pd external that implements the standard Schroeder-Moorer [45] reverb model. It uses eight comb filters and four all-pass filters on both left and right channel. The filter coefficients on left and right channels are slightly different to create a stereo effect.

A good thing in reverberation is that it is calculated for the whole outcoming signal, not for each sound source as the early reflections. Thus the computational load does not depend on the number of clappers. Moreover just the use of *freeverb˜* improves the results so much that there is no use for calculating the early reflections.

Chapter 5

Ensemble Control Modeling

Probably the most interesting part of this research is the ensemble modeling of several clappers. Each clapper can be thought as an oscillator that is connected to every other oscillator by the means of hearing. All the clappers in this population can clap individually without listening to other clappers or they can suddenly synchronize their clapping rate and start clapping in unison. The chaos can spontaneously transform to an order.

First references to this phenomenon of self-organization can be found from the 17th century when Dutch physicist Cristiaan Huygens found out that two pendulum clocks hanging next to another always synchronized their oscillation [48]. Probably the best known subject in the world of synchronization are the fireflies blinking on and off in unison in the riverbanks of Southeast Asia. The phenomenon of synchronization can be found everywhere in nature from atoms to planets. The laser beam comes from trillions of atoms pulsing in concert, all emitting photons of the same phase and frequency. In the heart about 10000 pacemaker cells called sinoatrial nodes are oscillating in unison to generate the electrical rhythm that sets the pace for heartbeat. The Moon turns on its axis at exactly the same rate as it orbits the Earth. More about the phenomenon of synchronization and self-organization can be read from the interesting book by Steven Strogatz [48].

5.1 Synchronization in Audience

5.1.1 Theories and Their Credibility

Néda and his colleagues have researched the humans' tendency to synchronize their clapping rates to a rhythmic applause. They have explained that the synchronization is achieved by the period doubling of the clapping rhythm [36, 37]. This feature does not exist in many other systems that are known to synchronize. Regardless of the period doubling, the results are understandable in the framework of the Kuramoto model of coupled nonlinear oscillators [30]. Two years later they introduced a new model of globally coupled two-mode stochastic oscillators [35]. It tries to overcome the shortages of their previous study. It takes into account the stochasticity of an individual oscillator and the coupling is realized through the global output of the system.

Another conflict in the Kuramoto model regarding to applause of a huge crowd is that it expects that all clappers are connected to each other equally. This is not the fact as a spectator of course hears the adjacent clappers better that those on the other side of a room. Yeung and Strogatz take this into account in their generalization of the Kuramoto model that allows time delayed interactions [55].

Common to these theories is that they are just generalizations of a mathematical model. Even though Kuramoto model is one of the most celebrated systems in nonlinear dynamics the problem is that no one knows if the model describes the reality faithfully. And this applies to all of the theories that try to explain this phenomenon of self-organization. There have been no tests so far because experiments are extremely difficult [48]. They require measurements of every individual oscillator. And these measurements have to be done for a whole network without violating the connections between oscilators so that the coupling process is not disturbed. It would have been very interesting to do some multichannel measurements for a small group of clappers to examine how are the clapping rates changed during synchronization. Unfortunately we did not have time for the measurements within the schedule of this research and they are left as a future plan.

However, we examined some live CD and DVD recordings. The result was one sample of applause with a synchronization that can be seen in Figure 5.1.

The synchronization does not appear very clearly in the absolute value of the signal (a) but the envelope can be brought out by temporal averaging (b). Some fluctuation can be seen in the envelope from the start of a clip. The synchronization becomes audible at roughly two seconds and it takes about two seconds to find a very clear synchronization. It is hard to say what causes the synchronization and why is it lost at the end. However, it looks like it takes some time to find a good synchronization but it can be lost a lot faster.

This little clip does not confirm nor refute the Kuramoto model. Only thing that can be derived from it, is that the OOI for synchronised clapping is roughly 400 ms.

5.2 Simulated Synchronization

5.2.1 Simple Model Based on Delay Lines

The first simulation of the synchronization phenomenon is quite simplified. Simplified block diagram of this control model can be seen in Figure 5.2 and a Pd patch in Figure 5.3. In fact the clappers are not modeled as separate oscillators as they should be. The level

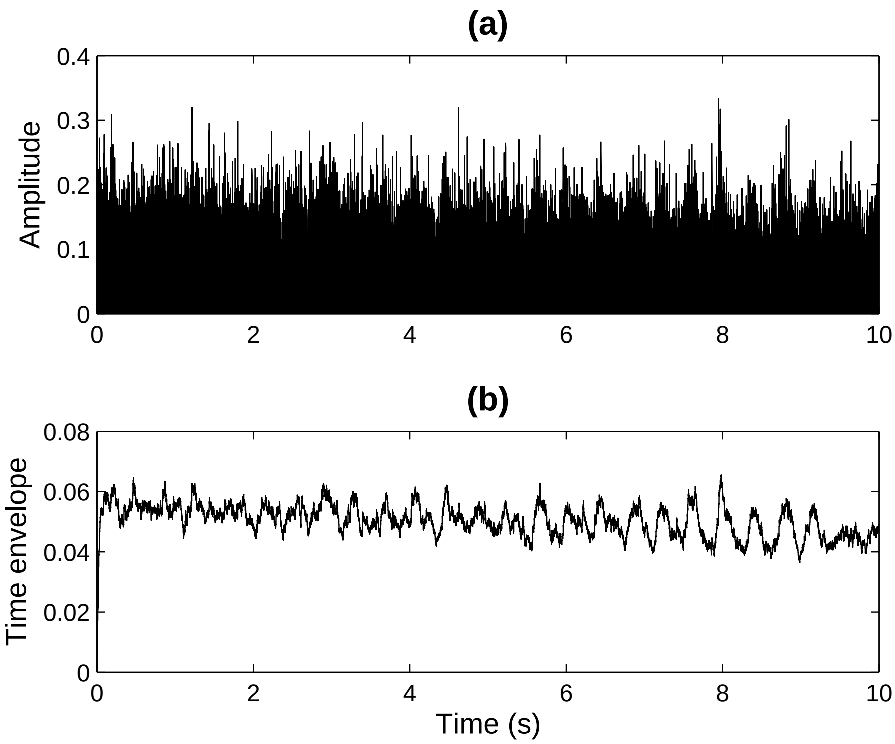

Figure 5.1: (a) Absolute value of a synchronized applause extracted from a live DVD recording. (b) Same signal with temporal averaging.

of synchronization is controlled with a slider. The model is based on a *metro* object that sends its bang messages to every clapper. As proposed by Néda et al. [36, 37] the clapping rate is slower when the synchronization is turned on. A clapper is modeled as a *pipe* object that can be seen inside the *clapper13* abstraction. The length of the delay line is randomly chosen to be between zero and the maximum OOI. If the synchronization level is raised, the length of a delay line is decreased. When a full synchronization is turned on, the length of a delay line is zero and all clappers are clapping in rhythm of the *metro*. Again when the synchronization is turned off, the delay lines become longer and all the clappers are clapping in different phases again.

Number of people in the audience can be controlled with a *ppl* external (see Appendix B.2.1). It does not actually turn on and off the clappers but rather blocks the bang messages that are triggering the synthesis model. Each *ppl* object gets its serial number as a creation argument. The number box that controls the number of clappers, sends its value to right inlet of *ppl* objects. *Ppl* passes the bang messages if the value is larger than or equal to the creation argument. Otherwise they are rejected

This simple model can be used if the number of clappers is relatively high (more than

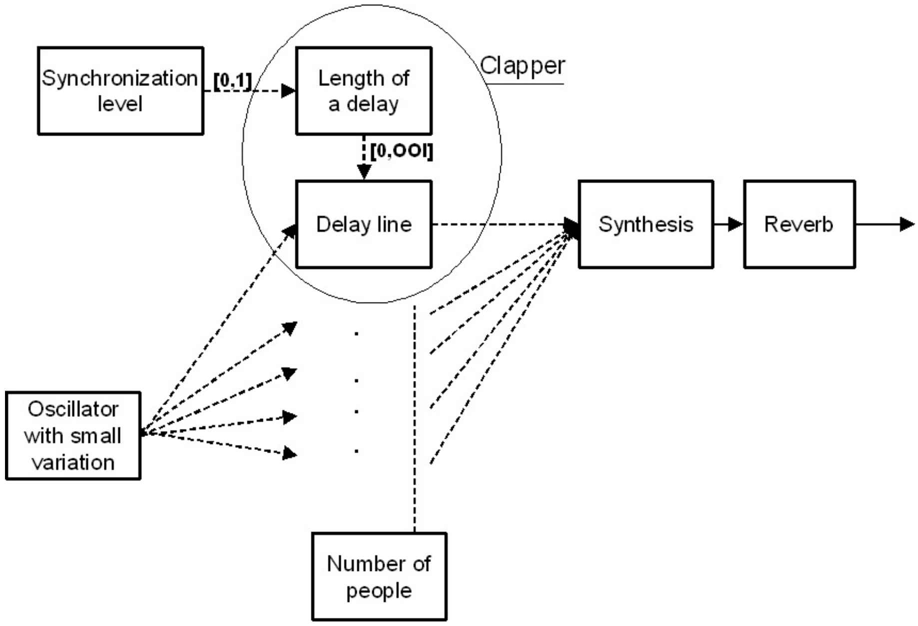

Figure 5.2: Simplified block diagram for a simple synchronization model.

20). With small number of clappers it is easily noticed that every clapper has the same clapping frequency and the result is not very realistic. With larger number of clappers this incompetence disappears. The synchronized clapping sounds quite realistic with this model even though it is so simple. This shows that good results can be obtained even if each clapper is not modeled as coupled oscillator.

5.2.2 A More Advanced Model: Coupled Oscillators

The second model is a lot more realistic. Simplified block diagram for this control model can be seen in Figure 5.4 and a Pd patch in a Figure 5.5. Every clapper is modeled as an individual oscillator. The fact that distinguishes this model from the Kuramoto model is that the oscillators do not listen to the global output of the system but to the lead synchronizer which is a *metro* object. Number of clappers can be controlled as it was controlled in previous model.

As Néda et al. proposed, every clapper has its natural clapping rate in the asynchronous mode and in synchronized mode the clapping rate is doubled. The distribution of natural clapping rates in not synchronized mode is twice as wide as it is in synchronized mode. These distributions can be seen in Figure 5.6 (a). Figure 5.6 (b) shows how the natural frequencies are distributed in my model. The clapping rates are sligthly faster as the ones proposed by Néda et al. [36, 37] because other measurements gave more faster clapping

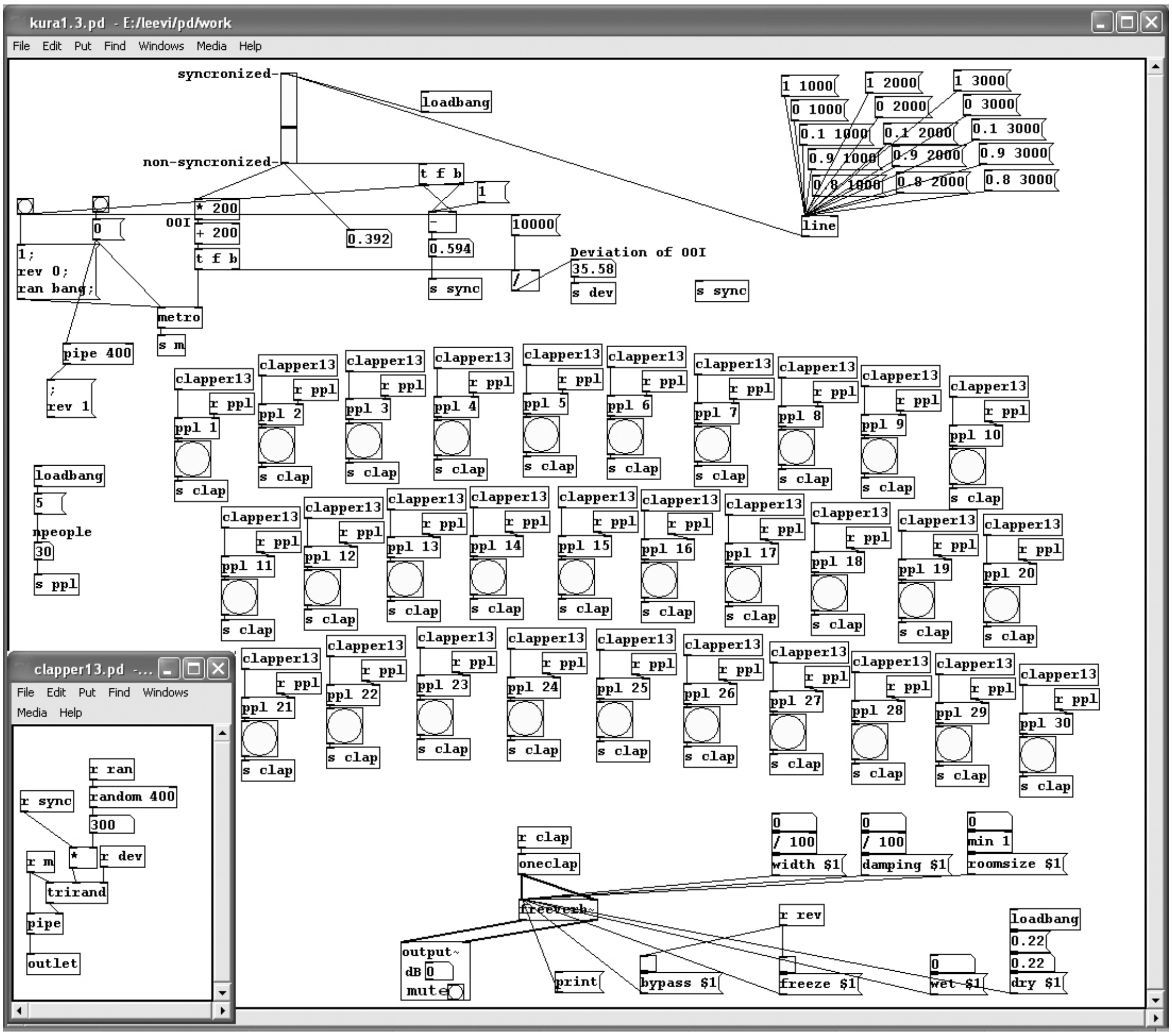

Figure 5.3: Pure Data patch for a simple synchronization model.

rates. While testing the final version, I also found out that the result sounds better if the clapping rates are sligthly faster. Equally every clapper has its own natural clapping rate when they are oscillating in the asynchronous mode. The clapping rates are assumed to have a triangular distribution to simplify the model. In synchronized mode every clapper aims to clap at the same rate as the *metro* object. This does not mean that everyone is clapping exactly at the same rate because a new OOI is calculated after every clap and it has some randomness.

The coupled oscillators are modeled as *cosc* objects. In Figure 5.5 they can be seen inside of the *clapper* abstraction. User can switch on and off the synchronization. There is also a slider with which the order level of the synchronization can be controlled. The clapping

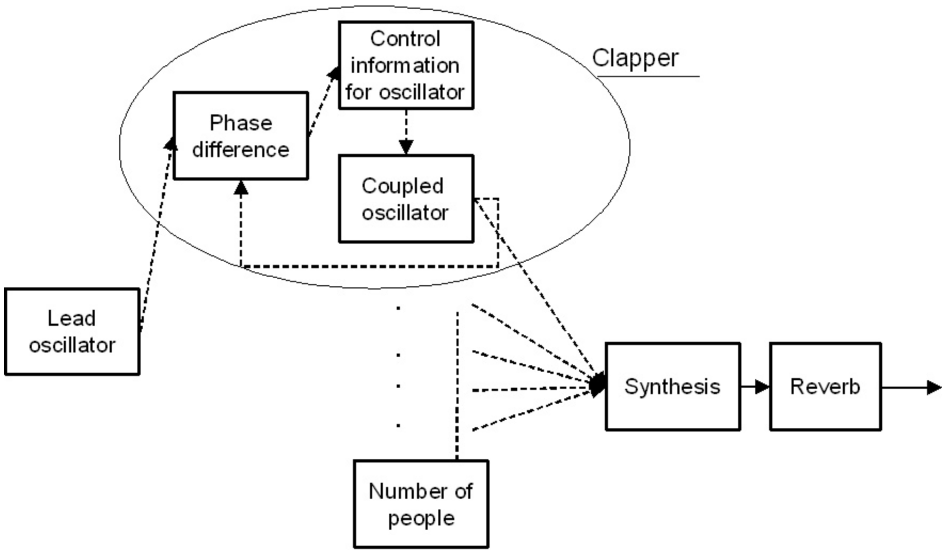

Figure 5.4: Simplified block diagram for a more advanced synchronization model that is based on coupled oscillators.

rate of an oscillator is controlled according to following rules:

1. In synchronized mode if the oscillator is trailing behind of the lead oscillator its clapping rate is accelerated. The amount of acceleration depends on how much it is delayed and what is the synchronization order.

2. Equally if the oscillator ahead of the lead oscillator its clapping rate is slowed down.

3. If the mode is switched to asynchronous, the clapping rate is gradually speeded up until the natural clapping frequency is achieved.

4. In both synchronized and asynchronous mode if the clapping rate is close enough for the preferred one, it is kept equal.

In addition to these rules there is always some randomness in the clapping rate. This is modeled so that every OOI has a 10% variation with triangular distribution.

The order coefficient determines how much the oscillator is allowed to have phase difference with the lead oscillator. It also sets the speed for the rate changing process. When the order is high the clapping is synchronized quickly and synchronization is very strong. Respectively when the order is low, it takes some time to find the synchronization and larger phase difference with the lead oscillator is allowed. The functionality of *cosc* external can be examined in more detail in Appendix B.2.2.

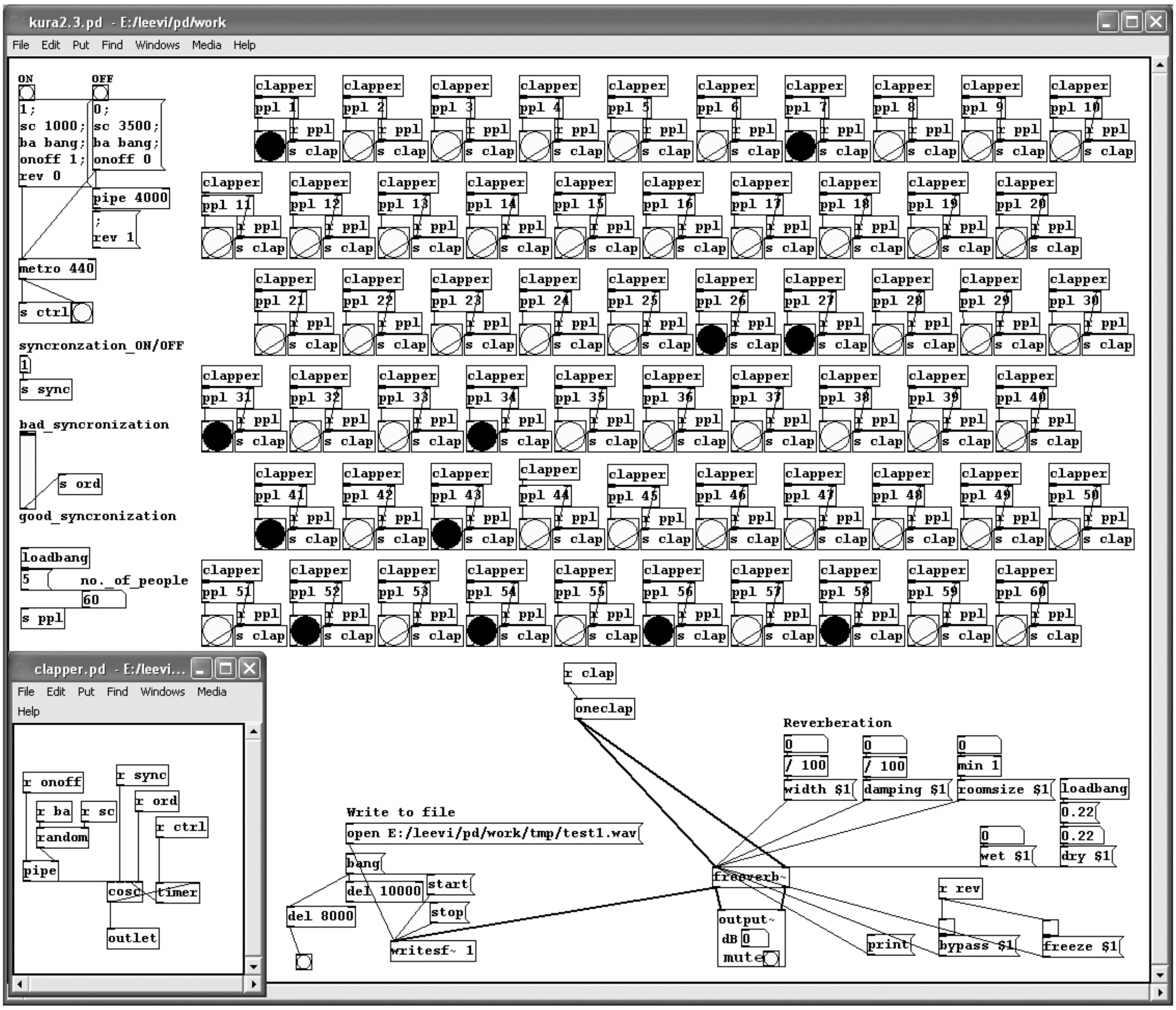

Figure 5.5: Pure Data patch for a more advanced synchronization model that is based on coupled oscillators.

Results obtained with these coupled oscillators are good. The process of finding synchronization and losing it sound very realistic. When only few clappers are modeled the order lever does not have so much influence but with larger crowds too accurate synchronization sounds quite unrealistic. Also too fast synchronization sounds quite unrealistic even though Kuramoto model implies that the synchronization should be reached without a delay. An example of a synthetic clapping sequence with synchronization can be seen in Figure 5.7. Signal (a) is a absolute value of a clapping sequence and (b) is the envelope of same signal gained by temporal averaging.

An important part of the model for many clappers is the reverberation. The same reverb

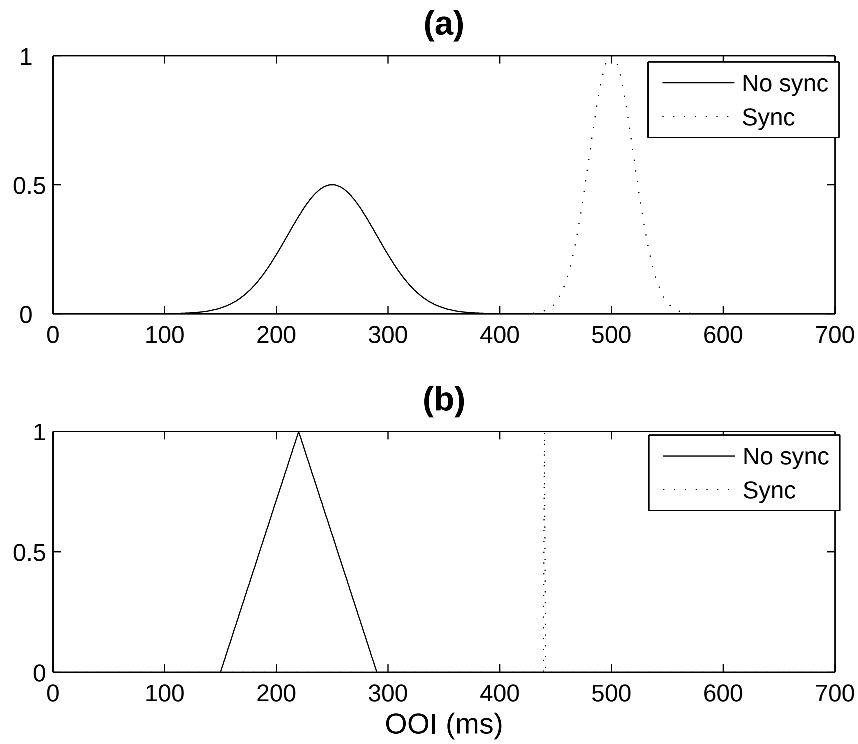

Figure 5.6: (a) Distribution of natural clapping rates as proposed by Néda et al. [36, 37]. (b) Distribution of natural clapping rates used in the coupled oscillators that are used to model clappers in an audience.

algorithm is used as in control model for single clapper but now it is in even bigger role. The result actually does not sound as an applause if no reverberation is used. The importance of reverberation was expected because of the fact that so big part of the perceived sound of hand clapping is just reverberation of the room.

The model of many synchronized clappers is computationally quite heavy. However this is not caused by the control model of coupled oscillators itself. As can be seen in Figure 5.5 there is a *bang* object for each clapper. It flashes every time a clapper claps. With 60 clappers this flashing takes about 50% of the CPU usage when the audio calculation takes only about 3%. The flashing is not obligatory for the model but it is a good visualization for the synchronization process and that is why it is kept in the model. If the flashing of clappers is removed the model is quite light and also bigger crowds can be modeled. Another big user of the CPU is the reverb algorithm but it is also very important for the quality of the sound and should not be removed.

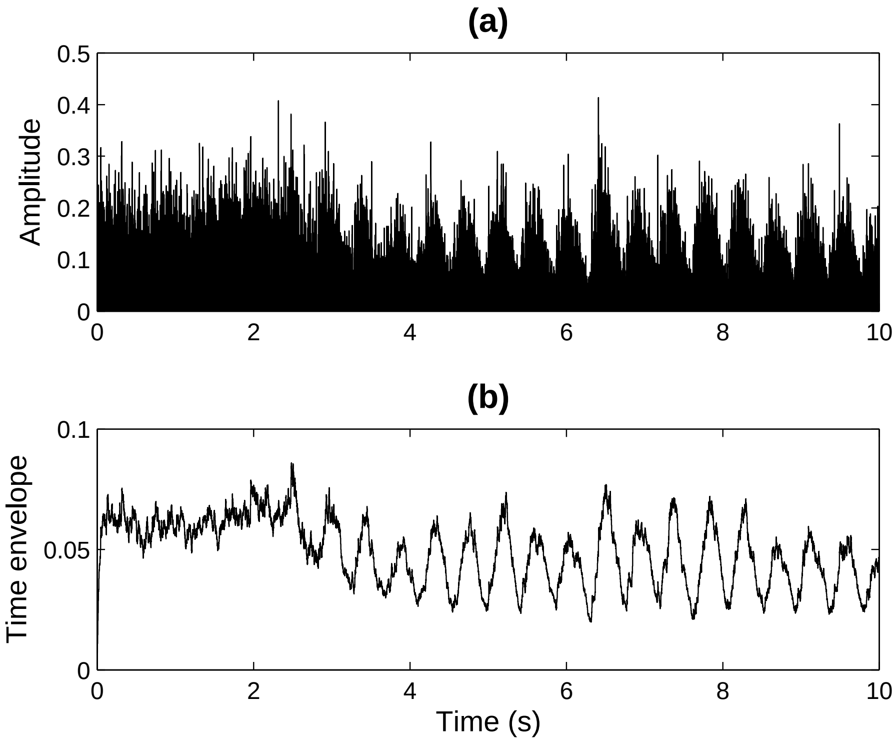

Figure 5.7: (a) Absolute value of a synchronized applause that was created using the control model that is based on coupled oscillators. (b) Same signal with temporal averaging.

Chapter 6

Conclusions and Future Work

In his chapter I will go through the results of this thesis and draw some conclusions of them. Both the synthesis model and the control model will be analyzed. I will also give some directions for possible future studies on this topic.

6.1 Synthesis Model

The first part of the research was to create a simplified but realistic synthesis model for a hand clap. The heart of this synthesis model is a two-pole resonator filter. Center frequency, bandwidth, and gain for this filter were derived from the measurements conducted in an anechoic chamber. Eight different clapping modes were used to get some information about the influence of alignment of hands. The strongest peak of the test claps was extracted using linear prediction.

The synthesis model is triggered with a Pd *bang* message. When a *bang* message is received, first the resonator coefficients are updated. This is done to give some variation to the produced sound. The variation of resonator coefficients is also based on the data derived from measurements. The resonator filter is excited with a short band-passed noise signal. The time envelope of the excitation signal is exponentially rising. The sound of a synthetic hand clap has two parts: The noisy attack part is when the excitation noise is fed into a resonator. This is followed by the decay that stands for a resonance of the cavity between hands.

The result of the synthesis is quite credible. However there are some shortcomings that could be improved or at least examined more carefully. One problem is that the noise envelope generator was first designed to create just isolated claps. When modeling multiple clappers with same synthesis model the control messages can arrive quite frequently. The problem is that if there already is a clap inside the system, it is discarded if a new clap comes in. This defect did not cause any problem when testing the ensemble control model for multiple clappers so it was left as it is.

Another problem came up when analyzing the data from measurements. We used three subjects in the measurements and they were asked to make a sequence of five claps in each of the eight clapping modes. As the sound of a hand clap is so stochastic there is a huge variation in the spectral shape even if the subject tries to maintain positioning of hands. Thus it is very difficult to derive the filter coefficients from only five claps in each mode. The test data should be a lot bigger in order to get more credible values for filter coefficients and their variations in different modes.

In general the synthesis model is quite successful. The model based on a single resonator that is excited with band-pass filtered noise pulses is computationally very light. Even lighter models could be defined using multirate signal processing. And as the sound event is so short and simple, there is no use for a heavier model. The model is also very flexible so it could be easily converted to other everyday sounds.

6.2 Control Models

The second part of the research focused on creating convincing control models for the synthesis. Two cases were implemented. The first one was the control model for a single clapper. It is based on 18 clapping sequences with different levels of enthusiasm. These 18 sequences were analyzed to find some patterns that are typical for humans when clapping hands. It was noticed that the variation of the clapping rate is not constant during the sequence. Especially at the start of a sequence the variation is larger. The clapping rate also had a tendency to fluctuate during longer sequences. Most remarkable this effect was in long enthusiastic clapping sequences where the clapping rate was slowing down before the end.

The control model that was implemented based on these observations is a good imitation of a clapping sequence of one person. On the other hand, just a small variation on the rate of a simple *metro* object would have created almost as realistic results. The problem is that it is quite hard to find use for this model of single clapper so I do not think there is need to investigate more.

The second more interesting case was to create a control model for several clappers. The emphasis was to model the synchronization phenomenon. First a quite simple model for synchronized applause was made. It consists of an oscillator that sends *bang* messages to clappers. Each clapper is a delay line with random length. When the synchronization level is raised, the delay lines get shorter and the phase of a clapper gets closer to the lead *bang* object. This model works quite well with larger crowds where the simplicity of the model disappears to the masses. However with small amount of people it is clearly heard that the clappers are clapping at the same rate and only the phase is different. Also the process of changing between synchronized and non synchronized applause needs to be improved.

An improvement was to create a model that is based on coupled oscillators. Every oscillator has its own preferred clapping rate. When the synchronization is turned on, the clappers start to change their clapping rates to find a synchronization with lead oscillator that is a *metro* object. When the synchronization is turned off, the clappers start again to clap with their own rates without listening to others. This model is more flexible for the number of clappers and the process of finding synchronization and losing it sounds realistic.

Even though the results of ensemble control modeling are satisfying there are lots of possibilities for future studies. The most interesting task would be to do some multi-channel measurements about the phenomenon. There are only mathematical models for the phenomenon of synchronization, but the validity of these models has never been tested because measurements are so difficult. For example, it is impossible to measure the blinking of individual fireflies without affecting the synchronization process at the same time. However, it would be possible to do these measurements for synchronized applause of small test crowds. One thing to examine in more detail is the process of finding synchronization and losing it. Based on these measurements, the model based on coupled oscillators could be made even more realistic.

Bibliography

[1] J.-M. Adrien. The missing link: modal synthesis. In G. De Poli, A. Picciali, and C. Roads, editors, *Representations of Musical Signals*, pages 269–297. The MIT Press, Massachusetts, 1991.

[2] P. Alku and T. Bäckström. Linear predictive method for improved spectral modelling of lower frequencies of speech with small prediction orders. *IEEE Transactions on Speech and Audio Processing*, 12(2):93–99, 2004.

[3] B. S. Atal and M. R. Schroeder. Predictive coding of speech signals. In *Proc. 1967 IEEE Conf. Communication Processing*, pages 360–361, 1967.

[4] F. Avanzini. Low-level models: resonators, interactions, surface textures. In D. Rocchesso and F. Fontana, editors, *The Sounding Object*, pages 137–172. Edizioni di Mondo Estremo, Firenze, Italy, 2003. Available online at http://www.soundobject.org/.

[5] F. Avanzini and D. Rocchesso. Efficiency, accuracy, and stability issues in discrete-time simulations of single reed wind instrument. *J. of the Acoustical Society of America*, 111(5):2293–2301, May 2002.

[6] D. R. Begault. *3-D Sound For Virtual Reality and Multimedia*. Academic Press, Cambridge, USA, 1994.

[7] R. Bresin and S. Dahl. Experiments on gestures: walking, running and hitting. In D. Rocchesso and F. Fontana, editors, *The Sounding Object*, pages 111–136. Edizioni di Mondo Estremo, Firenze, Italy, 2003. Available online at http://www.soundobject.org/.

[8] R. Bresin and A. Friberg. Emotional coloring of computer-controlled music performances. *Computer Music J.*, 24(4):44–63, 2000.

[9] R. Bresin, A. Friberg, and S. Dahl. Toward a new model for sound control. In *Proc. Conf. of Digital Audio Effects*, pages 45–49, Limerick, Ireland, 2001.

[10] C. Cadoz and J.-L. Florens. The physical model: modeling and simulating the instrument universe. In G. De Poli, A. Picciali, and C. Roads, editors, *Representations of Musical Signals*, pages 227–268. The MIT Press, Massachusetts, 1991.

[11] C. Carello, K.L. Anderson, and A.J. Kunkler-Peck. Perception of object length by sound. *Psychological Science*, 9:211–214, 1998.

[12] C. Carello, J. B. Wagman, and M. T. Turvey. Acoustic specification of object properties. In J. Anderson and B. Anderson, editors, *Moving Image Theory: Ecological Considerations*. Southern Illinois Press, 2003.

[13] PD community. Pd community site, May 2004. Available online at `http://pure-data.iem.at/about/`.

[14] P. R. Cook. Physically informed sonic modeling (PhISM): Synthesis of percussive sounds. *Computer Music J.*, 21(3):38–49, 1997.

[15] P. R. Cook. FOFs, wavelets, and particles. In *Real sound synthesis for interactive applications*, pages 149–168. A. K. Peters, Natick, Massachusetts, 2002.

[16] S. Dahl. The playing of an accent. Preliminary observations from temporal and kinematic analysis of percussionists. *J. of New Music Research*, 29(3):225–234, 2000.

[17] S. Dahl and R. Bresin. Is the player more influenced by the auditory than the tactile feedback from the instrument? In *Proc. Conf. of Digital Audio Effects*, pages 194–197, Limerick, 2001.

[18] F. Fontana. *Physics-based models for the acoustic representation of space in virtual environments*. PhD thesis, Univ. of Verona, 2003. Available online at `http://profs.sci.univr.it/~fontana/paper/20.pdf`.

[19] F. Fontana and R. Bresin. Physics-based sound synthesis and control: crushing, walking and running by crumpling sounds. In *Proceedings of the XIV Colloquium on Musical Informatics*, Firenze, Italy, 2003. Available online at `http://www.csis.ul.ie/dafx01/proceedings/index.htm`.

[20] D. J. Freed. Auditory correlates of perceived mallet hardness for a set of recorded percussive events. *J. of the Acoustical Society of America*, 87:311–322, January 1990.

[21] A. Friberg and J. Sundberg. Does music performance allude to locomotion? A model of final *ritardandi* derived from measurements of stopping runners. *J. of the Acoustical Society of America*, 105(3):1469–1484, March 1999.

[22] D. Gabor. Theory of communication. *J. of the Institute of Electrical Engineers Part III*, 93:429–457, 1946.

[23] D. Gabor. Acoustic quanta and the theory of hearing. *Nature*, 159:540–551, 1947.

[24] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects, part I. *J. Audio Eng. Soc.*, 19(6):462–470, 1971.

[25] L. Hiller and P. Ruiz. Synthesizing musical sounds by solving the wave equation for vibrating objects, part II. *J. Audio Eng. Soc.*, 19(7):542–550, 1971.

[26] A. Härmä and U. K. Laine. A comparison of warped and conventional linear predictive coding. *IEEE Transactions on Speech and Audio Processing*, 9(5):579–588, 2001.

[27] MIDI Manufacturers Association Incorporated. MIDI manufacturer's association, October 2004. Available online at `http://www.midi.org/`.

[28] M. Karjalainen, P. A. A. Esquef, P. Antsalo, A. Mäkivirta, and V. Välimäki. Frequency-zooming ARMA modeling of resonant and reverberant systems. *J. Audio Eng. Soc.*, 50(12):1012–1029, 2002.

[29] A. J. Kunkler-Peck and M. T. Turvey. Hearing shape. *J. of Experimental Psychology*, 26:279–294, 2000.

[30] Y. Kuramoto and I. Nishikawa. Statistical macrodynamics of large dynamical system. Case of plane transition in oscillator communities. *J. of Statistical Physics*, 49:569–605, 1987.

[31] S. Lakatos, S. McAdams, and R. Caussé. The representation of auditory source characteristics: simple geometric form. *Perception and Psychophysics*, 59:1180–1190, 1997.

[32] X. Li, R. J. Logan, and R. E. Pastore. Perception of acoustic source characteristics: Walking sounds. *J. of the Acoustical Society of America*, 90:3036–3049, December 1991.

[33] T. Lukkari and V. Välimäki. Modal synthesis of wind chime synthesis with stocastic event triggering. In *Proceedings of the 6th Nordic Signal Prosessing Symposium - NORSIG 2004*, pages 212–215, Espoo, Finland, June 2004. Available online at `http://wooster.hut.fi/publications/norsig2004/61_LUKKA.PDF`.

[34] O. Matthes. freeverb~, November 2004. Available online at http://www.akustische-kunst.de/puredata/.

[35] Z. Néda, A. Nikitin, and T. Vicsek. Synchronization of two-mode stochastic oscillators: a new model for rythmic applause an much more. *Physica A: Statistical Mechanics and its Applications*, 321:238–247, 2003.

[36] Z. Néda, E. Ravasz, Y. Brechet, T. Vicsek, and A.L. Barabási. Physics of the rhythmic applause. *Physical Review E*, 61:6987–6992, 2000.

[37] Z. Néda, E. Ravasz, Y. Brechet, T. Vicsek, and A.L. Barabási. The sound of many hands clapping. *Nature*, 403:849–850, 2000.

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

[39] B. H. Repp. The sound of two hands clapping: an exploratory study. *J. of the Acoustical Society of America*, 81:1100–1109, April 1987.

[40] C. Roads. Granular synthesis of sound. In C. Roads and J. Strawn, editors, *Fundations of Computer Music*, pages 145–159. The MIT Press, Massachusetts, 1985.

[41] C. Roads. Asynchronous granular synthesis. In G. De Poli, A. Picciali, and C. Roads, editors, *Representations of Musical Signals*, pages 143–185. The MIT Press, Massachusetts, 1991.

[42] C. Roads. *The Computer Music Tutorial*. The MIT Press, Massachusetts, 1995.

[43] D. Rocchesso, R. Bresin, and M. Fernström. Sounding objects. *IEEE Multimedia*, 10(2):42–52, 2003.

[44] X. Rodet, Y. Potard, and J.-B. Barrière. The CHANT project: from synthesis of singing voice to synthesis in general. *Computer Music J.*, 8(3):15–31, 1984.

[45] M. R. Schroeder. Natural sounding artificial reverbation. *J. Audio Eng. Soc.*, 10(3):219–224, 1962.

[46] J. O. Smith III. *Physical Audio Signal Processing: Digital Waveguide Modeling of Musical Instruments and Audio Effects, August 2004 Draft*, chapter Resonator Factoring. http://ccrma.stanford.edu/~jos/pasp/Resonator_Factoring.html, August 2004.

[47] K. Steiglitz. Feedback filters. In *A Digital Signal Processing Primer with Applications to Digital Audio and Computer Music*, pages 81–100. Addison-Wesley, 1996.

[48] S. Strogatz. *Sync: Rhythms of Nature, Rhythms of Ourselves*. Penguin Books, England, 2003.

[49] S. A. Tousman, R. E. Pastore, and S. Schwartz. Source characteristics: A study of hand clapping. *J. of the Acoustical Society of America*, 85:S53, 1989. Supplement 1.

[50] M. van Walstijn and M. Campbell. Discrete-time modeling of woodwind instrument bores using wave variables. *J. of the Acoustical Society of America*, 113(1):575–585, January 2003.

[51] V. Välimäki. Physics-based modeling of musical instruments. In *Proc. Stockholm Music Acoustics Conference*, pages 361–364, Stockholm, Sweden, August 2003.

[52] V. Välimäki. Physics-based modeling of musical instruments. *Acta Acustica united with Acustica*, 90:611–617, 2004.

[53] V. Välimäki, J. Huopaniemi, M. Karjalainen, and Z. Jánosy. Physical modeling of plucked string instruments with application to real-time sound synthesis. *J. Audio Eng. Soc.*, 44(5):331–353, 1996.

[54] W. H. Warren and R. R. Verbrugge. Auditory perception of breaking and bouncing events: A case study in ecological acoustics. *J. of Experimental Psychology: Human perception and Performance*, 10:704–712, 1984.

[55] S. Yeung and S. Strogatz. Time delay in the Kuramoto model of coupled oscillators. *Physica A: Statistical Mechanics and its Applications*, 321:238–247, 1998.

[56] U. Zölzer. *Digital Audio Signal Processing*, chapter Room Simulation, pages 181–205. Wiley, 1997.

Appendix A

Statistics from Measurements

In Table A.1 we can see the mean value and standard deviation of center frequency f_c, bandwidth B and gain g of two strongest resonances for all clapping modes. The statistics are counted from each clap of each subject (i.e. 15 claps in total for each mode). Table A.2 contains the same data for only authors hand claps (i.e. 5 claps in total for each mode).

Table A.1: The mean value and standard deviation of center frequency f_c, bandwidth B and gain g of the strongest resonance for all clapping modes with all subjects

	avg(f_c)	dev(f_c)	avg(B)	dev(B)	avg(g)	dev(g)
P1	1101 Hz	469 Hz	181 Hz	89 Hz	26,2	11,8
P2	846 Hz	412 Hz	195 Hz	129 Hz	16,8	6,4
P3	1505 Hz	350 Hz	280 Hz	116 Hz	22,8	10,1
A1	776 Hz	131 Hz	167 Hz	33 Hz	24,6	9,5
A2	1056 Hz	244 Hz	209 Hz	78 Hz	26,8	11,1
A3	1397 Hz	398 Hz	181 Hz	70 Hz	18,7	9,5
A1+	701 Hz	247 Hz	105 Hz	28 Hz	35,5	12,2
A1-	1037 Hz	340 Hz	246 Hz	162 Hz	14,2	3,2

Table A.2: The mean value and standard deviation of center frequency f_c, bandwidth B and gain g of strongest resonance for all clapping modes derived from only authors hand claps

	avg(f_c)	dev(f_c)	avg(B)	dev(B)	avg(g)	dev(g)
P1	1364 Hz	333 Hz	188 Hz	54 Hz	27,2	9,8
P2	774 Hz	448 Hz	139 Hz	40 Hz	13,8	3,5
P3	1562 Hz	257 Hz	375 Hz	94 Hz	19,5	6.3
A1	775 Hz	39 Hz	137 Hz	15 Hz	22,2	3.6
A2	1355 Hz	57 Hz	222 Hz	53 Hz	24,0	8.0
A3	1452 Hz	12 Hz	183 Hz	40 Hz	13,8	4.5
A1+	500 Hz	178 Hz	91 Hz	27 Hz	33,8	5,6
A1-	984 Hz	396 Hz	137 Hz	36 Hz	11,3	1,4

Appendix B

Implementation

This appendix introduces the source code of the Pd externals that are used in this work. First section describes the signal-externals that are used in the synthesis model of a single clap. Second section concentrates on a control externals that are used in the ensemble control modeling.

The source code as well as compiled Pd externals and Pd patches can be found at http://www.acoustics.hut.fi/publications/

B.1 Synthesis of a Single Clap

B.1.1 reson~.c

Reson~ external is the resonator from subsection 3.3.1 that is used in this hand clapping synthesis. It filters the input signal with a two-pole IIR-filter. Reson~ has inlets for center frequency (f_c), bandwidth (B), and gain (g). Every time when these inputs are changed, also the filter coefficients are updated.

```
#include "m_pd.h"
#include <math.h>

static t_class *reson_class;

typedef struct rctl {
  float c_y1; // old outputs for filtering
  float c_y2;
  float c_a0; // filter coefficients
  float c_a1;
  float c_a2;
} t_rctl;
```

```
typedef struct reson {
  t_object  x_obj;
  float x_b; // bandwidth
  float x_fc; // center frequency
  float x_g; // gain
  float x_r; // pole radius
  float x_theta; // pole angle
  float x_fs; // sampling rate
  t_rctl x_cspace; // links to
  t_rctl *x_ctl; // filter coefficients
  t_float x_f;
} t_reson;

static void reson_docoef(t_reson *x, t_floatarg fs, t_floatarg b, t_floatarg g);

/* constructor */
void *reson_new(t_floatarg fc, t_floatarg b, t_floatarg g) {
  t_reson *x = (t_reson *)pd_new(reson_class);
  inlet_new(&x->x_obj, &x->x_obj.ob_pd, gensym("float"), gensym("ft1"));
  inlet_new(&x->x_obj, &x->x_obj.ob_pd, gensym("float"), gensym("ft2"));
  inlet_new(&x->x_obj, &x->x_obj.ob_pd, gensym("float"), gensym("ft3"));
  outlet_new(&x->x_obj, gensym("signal"));

  x->x_fs = 44100;
  x->x_ctl = &x->x_cspace;
  x->x_cspace.c_y1 = 0;
  x->x_cspace.c_y2 = 0;
  reson_docoef(x,fc,b,g);
  x->x_f = 0;
  return (x);
}

/* update coefficients */
static void reson_docoef(t_reson *x, t_floatarg fc, t_floatarg b, t_floatarg g){
  x->x_fc=fc;
  x->x_b=b;
  x->x_g=g;
  x->x_r = 1 - x->x_b/x->x_fs*(float)3.14259;
  x->x_theta = (float)acos(2*x->x_r/(1+pow(x->x_r,2))
               *(float)cos(2*(3.14259f)*x->x_fc/x->x_fs));
  x->x_ctl->c_a0 = (1 - (float)pow(x->x_r,2))*(float)sin(x->x_theta)*x->x_g/2;
  x->x_ctl->c_a1 = 2*x->x_r*(float)cos(x->x_theta);
  x->x_ctl->c_a2 = -1*(float)pow(x->x_r,2);
}

static void reson_ft1(t_reson *x, t_floatarg fc){
```

```
  reson_docoef(x, fc, x->x_b, x->x_g);
}

static void reson_ft2(t_reson *x, t_floatarg b){
  reson_docoef(x, x->x_fc, b, x->x_g);
}

static void reson_ft3(t_reson *x, t_floatarg g){
  reson_docoef(x, x->x_fc, x->x_b, g);
}

t_int *reson_perform(t_int *w) {
  float *in1 = (float *)(w[1]);
  float *out = (float *)(w[2]);
  t_rctl *c = (t_rctl *)(w[3]);
  int n = (t_int)(w[4]);
  int i;
  float a0 = c->c_a0;
  float a1 = c->c_a1;
  float a2 = c->c_a2;
  float y1 = c->c_y1;
  float y2 = c->c_y2;

  for (i = 0; i < n; i++) {
    float y0 = a0*(*in1++) + a1*y1 + a2*y2;
    *out++ = y0;
    y2=y1;
    y1=y0;
  }
  c->c_y1=y1;
  c->c_y2=y2;
  return (w+5);
}

void reson_dsp(t_reson *x, t_signal **sp) {
  x->x_fs = sp[0]->s_sr;
  reson_docoef(x, x->x_fc, x->x_b, x->x_g);
  dsp_add(reson_perform, 4, sp[0]->s_vec, sp[1]->s_vec, x->x_ctl, sp[0]->s_n);
}

void reson_tilde_setup(void) {
  reson_class = class_new(gensym("reson~"),
           (t_newmethod)reson_new,
              0, sizeof(t_reson),
           0, A_DEFFLOAT, A_DEFFLOAT, A_DEFFLOAT, 0);
  class_addmethod(reson_class,(t_method)reson_dsp, gensym("dsp"), 0);
```

```
  class_addmethod(reson_class, (t_method)reson_ft1, gensym("ft1"), A_FLOAT, 0);
  class_addmethod(reson_class, (t_method)reson_ft2, gensym("ft2"), A_FLOAT, 0);
  class_addmethod(reson_class, (t_method)reson_ft3, gensym("ft3"), A_FLOAT, 0);
  CLASS_MAINSIGNALIN(reson_class, t_reson, x_f);
}
```

B.1.2 noiseq˜.c

Noiseq˜ is a noise envelope generator. It takes noise signal as an input and when a bang message is received a noise pulse described in subsection 3.3.2 is send out. When a bang message comes in, noiseq˜ also checks if there is already another noise pulse inside and warns reson˜ so that the filter coefficients are not changed in the middle of a clap. Second inlet is used to switch on and off the exponential decay of a noise pulse (recommended off).

```
#include "m_pd.h"

static t_class *noiseq_class;

typedef struct _noiseq {
  t_object  x_obj;
  t_int x_count;    //counts samples after a bang
  t_float x_donoff; //switch on/off the decaying part of the excitation signal
  t_float x_env;    //show if there is already a clap inside when a new one comes.
                    //(used to decide if the reson~ coefficients can be changed)
  t_float x_namp;   //amplitude of the noise excitation
  t_float x_decay;  //constant for exponential decay/rise
  t_outlet *f_out;
  t_sample x_f;
} t_noiseq;

/* restart counter and increase noise amplitude */
void noiseq_bang(t_noiseq *x) {
  if (x->x_count > 140)
    x->x_namp = (0.02f);
  else{
    x->x_namp = (0.03f);
  }
  x->x_count=0;
  outlet_float(x->f_out, x->x_env);
}

void *noiseq_new(t_floatarg f) {
  t_noiseq *x = (t_noiseq *)pd_new(noiseq_class);
  inlet_new(&x->x_obj, &x->x_obj.ob_pd, gensym("float"), gensym("onoff"));
  x->x_count=0;
```

```
  x->x_donoff=0;
  x->x_decay=(float)0.99;
  x->x_namp=0;
  x->x_env=1;
  outlet_new(&x->x_obj, gensym("signal"));
  x->f_out = outlet_new(&x->x_obj, &s_float);
  return (void *)x;
}

static void noiseq_onoff(t_noiseq *x, t_floatarg donoff){
  x->x_donoff=donoff;
}

t_int *noiseq_perform(t_int *w) {
  t_float *in = (t_float *)(w[1]);
  t_float *out = (t_float *)(w[2]);
  t_noiseq *x = (t_noiseq *)(w[3]);
  int n = (int)(w[4]);
  int i;
  /* increase or decrease noise amplitude namp and multiply input with it */
  while(n--) {
    float f = *(in++);
    f=f*x->x_namp;
    *out++ = f;

    if (x->x_count < 140) {
      x->x_count++;
      x->x_env = 0;
      x->x_namp = x->x_namp / x->x_decay;
    }
    else if (x->x_count < 600) {
      x->x_count++;
      x->x_env = 0;
      x->x_namp = x->x_namp * x->x_decay *x->x_donoff;
    }
    else{
      x->x_env = 1;
      x->x_namp = x->x_namp * x->x_decay *x->x_donoff;
    }
  }
  return (w+5);
}

void noiseq_dsp(t_noiseq *x, t_signal **sp) {
  dsp_add(noiseq_perform, 4, sp[0]->s_vec, sp[1]->s_vec, x, sp[0]->s_n);
}
```

```
void noiseq_tilde_setup(void) {
  noiseq_class = class_new(gensym("noiseq~"), (t_newmethod)noiseq_new,
                           class_addmethod(noiseq_class,(t_method)noiseq_dsp,
                           gensym("dsp"), 0);
  class_addmethod(noiseq_class, (t_method)noiseq_onoff, gensym("onoff"), A_FLOAT, 0);
  CLASS_MAINSIGNALIN(noiseq_class, t_noiseq, x_f);
  class_addbang(noiseq_class, noiseq_bang);
}
```

B.1.3 twopz˜.c

Twopz˜ is a slightly modified version of biquad˜ that is included in Pd [13]. It is a two-pole two-zero filter that is used in the inverse filtering of a signal as described in subsection 3.2.3.

B.2 Ensemble Control Model

B.2.1 ppl.c

Ppl is a control object that is used to control the number of clappers. It passes incoming bang messages if the right inlet is larger than or equal to the creation argument.

```
#include "m_pd.h"
#include <string.h>
#include <stdio.h>

t_class *ppl_class;

typedef struct _ppl {
  t_object x_obj;
  float x_state;
  float x_no;
} t_ppl;

void *ppl_new(t_floatarg f) {
  t_ppl *x = (t_ppl *)pd_new(ppl_class);
  floatinlet_new(&x->x_obj, &x->x_state);
  outlet_new(&x->x_obj, 0);
  x->x_state = 0;
  x->x_no = f;
  return (x);
}

void ppl_bang(t_ppl *x) {
```

```
  if (x->x_state >= x->x_no) outlet_bang(x->x_obj.ob_outlet);
}

void ppl_float(t_ppl *x, t_float f) {
  if (x->x_state >= x->x_no) outlet_float(x->x_obj.ob_outlet, f);
}

void ppl_setup(void) {
  ppl_class = class_new(gensym("ppl"), (t_newmethod)ppl_new, 0,
  sizeof(t_ppl), 0, A_DEFFLOAT, 0);
  class_addbang(ppl_class, ppl_bang);
  class_addfloat(ppl_class, ppl_float);
}
```

B.2.2 cosc.c

Cosc is a coupled oscillator which synchronizes the rhythm to the lead oscillator if the sychronization is turned on. In asynchronous mode cosc is oscillating with its own natural rate.

```
#include "m_pd.h"
#include <stdio.h>
#include <stdlib.h>

t_class *cosc_class;

typedef struct _cosc {
  t_object x_obj;
  t_clock *x_clock;
  int x_hit;
  double x_deltime;    //holds the currend preferred clapping rate
  float x_delay;       //current delay until the next clap
  t_float x_pshift;    //phase difference with leader oscilator
  t_float x_ooi_sync;  //preferred clapping rate in synchronizated state
  t_float x_ooi_enth;  //preferred clapping rate in unsynchronizated state
  t_float x_sync;      //synchronization on/off
  t_float x_order;     //parameter that describes how syncronizated the audience is
} t_cosc;

void cosc_sync(t_cosc *x, t_floatarg sync);
void cosc_order(t_cosc *x, t_floatarg order);
void cosc_tick(t_cosc *x);

void *cosc_new(t_floatarg f) {
  t_cosc *x = (t_cosc *)pd_new(cosc_class);
```

```
  outlet_new(&x->x_obj, gensym("bang"));
  inlet_new(&x->x_obj, &x->x_obj.ob_pd, gensym("float"), gensym("sync"));
  inlet_new(&x->x_obj, &x->x_obj.ob_pd, gensym("float"), gensym("pshift"));
  inlet_new(&x->x_obj, &x->x_obj.ob_pd, gensym("float"), gensym("order"));

  x->x_pshift = 0;;
  x->x_clock = clock_new(x, (t_method)cosc_tick);
  x->x_ooi_sync = 440;
  //triangular distribution with average of 220
  x->x_ooi_enth = 150 + 70*rand()/RAND_MAX + 70*rand()/RAND_MAX;
  cosc_sync(x, 0);
  cosc_order(x, 0);
  x->x_hit = 0;
  return (x);
}

void cosc_tick(t_cosc *x) {
  x->x_hit = 0;
  outlet_bang(x->x_obj.ob_outlet);

  //in syncronized mode, speed up the clapping rate if the oscilator
  //is trailing behind the control oscilator
  if (x->x_pshift < 220 && x->x_pshift > (30+70*x->x_order) && x->x_sync == 1){
    x->x_delay = (x->x_delay*x->x_order + x->x_ooi_sync*(1-x->x_order) + x->x_ooi_sync)/2
    - x->x_pshift/(3+4*x->x_order);
  }
  //in syncronized mode, slow down the clapping rate if the oscilator
  //is ahead of the control oscilator
  else if (x->x_pshift < (420-70*x->x_order) && x->x_pshift >= 220 && x->x_sync == 1){
    x->x_delay = (x->x_delay*x->x_order + x->x_ooi_sync*(1-x->x_order) + x->x_ooi_sync)/2
    + (x->x_ooi_sync - x->x_pshift)/(3+4*x->x_order);
  }
  //when changing to the unsyncronized mode, speed up the clapping rate
  //until the own clapping rate is reached
  else if (x->x_sync == 0 && x->x_delay > x->x_deltime*(1.05+0.25*x->x_order)){
    x->x_delay = x->x_delay/(1.3-0.25*x->x_order);
  }
  //if everything is ok, just keep on clapping
  else {
    x->x_delay = x->x_deltime;
  }

  //waiting time before the next clap has 10% triangular distribution
  x->x_delay = 0.9*x->x_delay + 0.1*x->x_delay*rand()/RAND_MAX +
               0.1*x->x_delay*rand()/RAND_MAX;
  if (!x->x_hit) clock_delay(x->x_clock, x->x_delay);
```

```
}

void cosc_float(t_cosc *x, t_float f) {
  if (f != 0) cosc_tick(x);
  else clock_unset(x->x_clock);
  x->x_hit = 1;
}

void cosc_bang(t_cosc *x) {
  cosc_float(x, 1);
}

void cosc_stop(t_cosc *x) {
  cosc_float(x, 0);
}

void cosc_sync(t_cosc *x, t_floatarg sync){
  x->x_sync=sync;
  if (x->x_sync == 1) x->x_deltime = x->x_ooi_sync;
  else x->x_deltime = x->x_ooi_enth;
}

void cosc_order(t_cosc *x, t_floatarg order){
  x->x_order=order;
}

void cosc_pshift(t_cosc *x, t_floatarg pshift){
  x->x_pshift=pshift;
}

void cosc_free(t_cosc *x) {
  clock_free(x->x_clock);
}

void cosc_setup(void) {
  cosc_class = class_new(gensym("cosc"), (t_newmethod)cosc_new,
     (t_method)cosc_free, sizeof(t_cosc), 0, A_DEFFLOAT, 0);
  class_addbang(cosc_class, cosc_bang);
  class_addmethod(cosc_class, (t_method)cosc_stop, gensym("stop"), 0);
  class_addmethod(cosc_class, (t_method)cosc_sync, gensym("sync"), A_FLOAT, 0);
  class_addmethod(cosc_class, (t_method)cosc_order, gensym("order"), A_FLOAT, 0);
  class_addmethod(cosc_class, (t_method)cosc_pshift, gensym("pshift"), A_FLOAT, 0);
  class_addfloat(cosc_class, (t_method)cosc_float);
}
```