Physically Informed Sonic Modeling (PhISM): Percussive Synthesis

Perry R. Cook
Department of Computer Science,
Department of Music, Princeton University
prc@cs.princeton.edu

Abstract

This paper describes methods for analysis/synthesis of percussive sounds, applying principles from physical modeling, and coupling the results to parametric synthesis techniques such as modal, sinusoidal, and granular synthesis. Emphasis is placed upon using physical modeling to provide expressive control of efficient synthesis models. Examples presented include marimba, police whistle, and maracha.

1 Introduction

As synthesis using physical modeling becomes more widespread and works its way into mainstream commercial workstation software and dedicated music synthesis hardware, large subsets of traditional instruments have yet to be explored using physical models. This paper focuses on natural and expressive synthesis of sounds in the percussion family, and describes a set of methods for systematically analyzing and parametrically synthesizing many percussion instruments. Since percussion instruments arguably represent a broader spectrum of physical configurations than any other instrument family (bars, plates, membranes, cavity and tube resonators, and non-linearites of countless varieties, all coupled to each other in varieties of ways), there exist no common model features which allow the percussion family to be captured with simple variations in model topology or variations of the parameters of a meta-model. By grouping the percussion family into sub-families, some similarities can be exploited, but there are still large varieties of configurations and excitation means. The expressive nature of percussion lies in the ability to excite (usually strike) objects, with a variety of objects, in a variety of locations, and in a variety of ways. Sampling synthesis based on digital recordings lacks the obvious means for manipulating meaningful parameters in musical ways. Direct real-time synthesis of many percussion instruments by physical modeling is likely to remain beyond practical computing means for some time to come. Further, the physical mechanisms of many such systems are still not completely understood.

The analysis/synthesis approach described here, called Physically Informed Sonic Modeling (PhISM), draws upon many of the techniques found in research in physical modeling, Fourier analysis/synthesis, wavelet analysis/synthesis, and granular synthesis. The primary goal is to provide expressive synthesis, with meaningful control parameters similar to those a percussionist would use to create subtle expression on an actual instrument. Secondary goals are efficiency of the algorithms, and the ability of the system to behave somewhat as a traditional analysis/synthesis technique. Motivated by the two distinct families of percussion instruments modeled in this paper, two PhISM-family synthesis algorithms will be presented. The first algorithm, called Physically Informed Spectral Additive Modeling (PhISAM), is implemented using modal synthesis (resonant filters). PhISAM is suitable for bells, bar percussion, and other such instruments exhibiting exponentially decaying resonant behaviors. The second synthesis algorithm, called Physically Informed Stochastic Event Modeling (PhISEM), is based on pseudo random overlapping and adding of small grains of sound according to predetermined physical rules and parameters. The PhISEM algorithm family is suitable for instruments such as maracha, guiro, tambourine, and others characterized by random interactions of sound-producing component objects. Both model families allow for scaleable amounts of modeling of the physics of the instrument and excitation. One of many possible synthesis techniques can be employed for synthesizing the actual sound, such as amplitude modulated oscillator synthesis, frequency modulation, stochastically parameterized sampling (granular) synthesis, and modal synthesis.

2 PhISAM: Physically Informed Spectral Additive Modeling

The PhISAM (Physically Informed Spectral Additive Modeling) algorithm is based on modal synthesis, but could be realized with sinusoidal oscillators as well. The modal filter or oscillator control parameters are driven and controlled by rules derived from predetermined Fourier boundary methods, and/or from analysis data extracted from recorded sounds. The PhISAM algorithm is suitable for resonant percussion instruments, such as marimba, xylophone, vibraphone, cowbell, agogo, and other instruments characterized by impulsive excitation of a relatively few exponentially decaying, weakly coupled sinusoidal modes. Some researchers have synthesized sound directly from finite difference solutions of the physics of bar percussion instruments such as the xylophone [Doutaut and Chaigne 1993], but the purpose of this project is to achieve efficient and expressive synthesis, wherever possible applying knowledge derived from physical modeling. Serra [1986] proposed synthesis of the marimba using sinusoidal modeling with added residual noise, and Wawrzynek [1989] proposed a multiple filter model of the marimba with impulsive and noise excitation. PhISAM affords the same flexible control over the modal parameters described by Serra and Wawrzynek, but also provides expressive yet simple control over excitation parameters such as stick hardness, and important performance parameters such as strike position and vigor of the strike. Such physical hooks allow additional modeling of the performer, allowing for implementation of occasional accidental multiple strikes and missed strikes.

Modes of vibration of percussion instruments can be derived from acoustical theory [Moore 1970], by modal analyses of physical systems [Rossing 1976], by peak-picking from spectra calculated by Fourier Analysis [Serra and Smith 1990], adaptive filter analysis such as LPC [Serra 1986], or other all-pole filter modeling techniques [Larouche and Meillier 1994]. Spectrum analysis methods are in many ways most desirable, because new systems can be treated rapidly with signal processing techniques applied to recordings. Theory can be developed as needed or convenient to add heuristics about the behavior of modes. By using a high order filter analysis the modes of significance can be detected, and weak modes can be discarded, or included as part of the excitation. If an adaptive technique is used for LPC or ARMA modeling, it is often advantageous to perform the analysis backwards in time, progressing in the analyzed signal from more pitched to non-pitched.

A prototype excitation is constructed either by recording excitations in a non-resonant instrument condition (damping the bar of a vibraphone and recording an actual stick strike, for example), or by extracting a residual resulting from inverse filtering or Fourier-based deterministic/stochastic decomposition [Serra and Smith 1990]. Once derived, the prototype excitation can then be manipulated in the time domain (filtering, stretching by interpolation, etc.) and/or frequency domain to yield a flexible control space for excitation wave functions. A derived family of parametrically controlled excitation pulses can be related to (and realized by) wavelet analysis/synthesis, with a stochastic component representing the noisy character of percussive excitations.

To simulate strike location, simple rules can be derived to control the relative levels of each mode in the modal synthesis, based on the spatial vibration patterns of each mode. Strike position data can also be inferred from modal analysis, or Fourier analysis of actual strikes in multiple positions. Some randomness can also be included in the resynthesis rule process. The C++ code of Example 1 demonstrates a very simple but effective mapping of strike position to synthesis parameters. Note that if the filters are implemented in parallel, and the individual mode gains are applied to the filter inputs, continuity is maintained in modes which are still ringing, and new excitations interact naturally with current energy in the system.

```
void Marimba :: setStrikePos(float pos)  {
   float temp;
   temp = sin(1.0 * pos * PI);
   this->setFiltGain(0,0.12 * temp);      /*  1st mode */
   temp = sin(0.03 + 3.9 * pos * PI);
   this->setFiltGain(1,0.02 * temp);      /*  2nd mode*/
   temp = sin(0.05 + 9.3 * pos * PI);
   this->setFiltGain(2,0.03 * temp);      /*  3rd mode */
                                  /* Leave 4th mode alone  */ }
```

C++ Example 1: Marimba strike position (0.0 and 1.0 at each end of the bar, 0.5 at the center) affects excitation level of the first three modes.

For further performance realism, simple physics of the percussion mallets and the performer can be included [Garton 1992][Jànosy et. al. 1994]. The resynthesis model can be built and implemented in a modular fashion, using an expert/player/instrument structure as described in [Cook 1995]. Figure 1 shows a typical analysis/resynthesis process schematically, including spectrograms of an original and resynthesized marimba tone, and the inclusion of player dynamics and constraints in controlling the marimba model.

3 PhISEM: Physically Informed Stochastic Event Modeling

The PhISEM (Physically Informed Stochastic Event Modeling) algorithm is based on pseudo-random overlapping and adding of small grains of sound, or pseudo-random modification of the parameters of a parametric synthesis model, according to rules and parameters derived from off-line physical simulations and heuristics. The PhISEM algorithm is suitable for instruments such as maracha, guiro, police whistle, etc., where sound is generated and/or modified by a

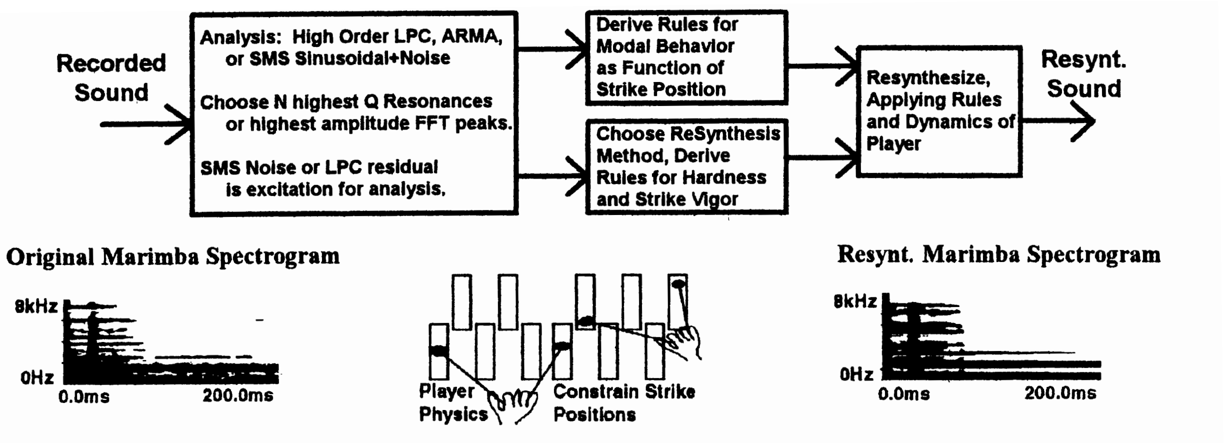
Recorded Sound
Analysis: High Order LPC, ARMA, or SMS Sinusoidal+Noise
Choose N highest Q Resonances or highest amplitude FFT peaks.
SMS Noise or LPC residual is excitation for analysis.
Derive Rules for Modal Behavior as Function of Strike Position
Choose ReSynthesis Method, Derive Rules for Hardness and Strike Vigor
Resynthesize, Applying Rules and Dynamics of Player
Resynt. Sound
Original Marimba Spectrogram
8kHz
0Hz
0.0ms
200.0ms
Player Physics
Constrain Strike Positions
Resynt. Marimba Spectrogram
8kHz
0Hz
0.0ms
200.0ms

Figure 1 PhISAM Analysis/Synthesis for a marimba. Analyzed original and resynthesis spectrograms of a marimba attack are shown. The lower center block represents performance rules based on player constraints.

characteristic discrete event, such as the pea inside a police whistle crossing the airway entrance, or a bean striking the inside surface of a maracha gourd. Such systems can be simulated to control synthesis which models discrete events directly, or can be used for collecting and controlling statistics which change as a function of the excitation and system parameters to control an efficient parametric synthesis model.

At the heart of PhISEM algorithms are particle models, characterized by basic Newtonian equations governing the motion and collisions of point masses as can be found in any introductory physics textbook. These systems can be solved numerically [Gould and Tobochnik 1996] using simple forward or backward Euler difference methods. In the simple Euler method, velocity is approximated by the difference between the current discrete time position and the last, divided by the time step Δt. Similarly the acceleration is approximated by the first difference of velocity.

$$velocity = v(t) = \frac{\partial x(t)}{\partial t} \approx \frac{x(t) - x(t-\Delta t)}{\Delta t}$$

$$acceleration = a(t) = \frac{\partial v(t)}{\partial t} \approx \frac{x(t) - 2x(t-\Delta t) + x(t-2\Delta t)}{\Delta t^2}$$

By selecting a time step which is suitably small, appropriate accuracy and stability can be achieved. These equations become vectors in two and three dimensions. At collisions, momentum and center of mass are used for additional solution constraints. Runga Kutta methods can be used for more accuracy at larger time steps.

As a first example, a simple two-dimensional single particle model can derived for a police/referee whistle. Figure 2 shows a side view of the whistle, with a single particle 'pea' suspended in a breath-controlled vector field. The region around the fipple where oscillation takes place is characterized by a random vector field, reflecting turbulence and vortices.

One might be tempted to connect the particle model parameters to an accurate physical model of the air jet fluid dynamics [Verge 1995][Chafe 1995], but since the goal of this project is to couple physical simulations to efficient synthesis techniques, the sound of the police whistle can be captured well by using data from the vector field and position of the pea to control a simple oscillator. Experiments and spectrograms using a real police whistle show a decrease in pitch, but an increase in amplitude whenever the pea is in the immediate region of the jet oscillator. These effects are easily coupled from the parameters of a physical simulation to the control inputs of a simple oscillator synthesis model as shown in Figure 2. This allows the entire model to run easily in real-time under user controlled breath pressure.

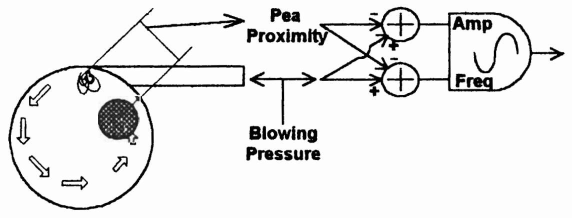

Figure 2: Whistle model showing physical control of oscillator synthesis. Vector flow field is shown pushing pea into turbulence at entrance.

The particle model can be extended to multiple-particle systems in three dimensions, such as the maracha or other shaker instruments. Calculating these models in real time may not be practical with current computing equipment, but off-line simulations using particle systems can be used for calculating time-varying statistical distributions, to be used in real time to control a parametric synthesis algorithm. A spherical maracha model containing 25 "beans" was constructed, virtually "shaken" with various velocity parameters, and statistics were collected of the frequency, waiting time, and sound intensity of exterior collisions. Only beans hitting the outside shell of the maracha were considered significant sound-producing collisions, since bean-bean collisions inside the gourd do not couple efficiently to the radiated sound. The square of the velocity normal relative to the maracha surface was used to calculate the strength of the radiated sound, and corresponding energy was taken out of the colliding bean's velocity at the time of a maracha wall collision. Bean-bean collisions also caused a decrease in energy. As might be expected, once energy is put into the system, the intensity of collisions decays exponentially. The likelihood of a significant sound producing collision occurring was found to be roughly constant, except when all beans are nearly at rest. Figure 3 shows a Poisson distribution of waiting time until the next collision, collected by simulating the shake of a 25 bean maracha model ten times and forming a histogram of the waiting times.

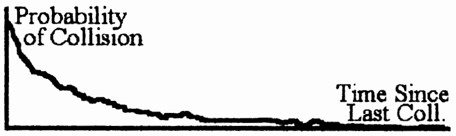

Figure 3. Histogram of waiting time until next sound-producing event for maracha model. Values of axes are a function of the gourd size and number of beans.

Once statistics are collected, parametric random number distributions which model the simulated distributions can be derived, and used to control the overlapping and adding of a number of single-collision sounds. These sounds could be acquired by recording single beans dropping onto an actual maracha, or synthesized using a parametric algorithm. For this project, an efficient maracha synthesis model was developed using rapidly decaying noise for the individual collision events, and a single biquad filter to model the resonance of the maracha gourd. Since the sum of exponentially decaying random noises is equal to a single noise source multiplied by an exponentially decaying value which is increased additively by each new collision, only a single exponential decay and a single noise source is required to compute the total sound. A slower exponential decay to model net system energy controls the event statistics. Requiring two random number generators for collision event likelihood and sound, two exponential decays, and one biquad filter, this model is approximately the same complexity as synthesis using interpolated PCM wavetables and filtering. The flexibility of real-time manipulation of control statistics and gourd resonance, however, makes the model much more playable in an expressive sense, and makes the model reconfigurable to a large variety of instruments of the shaker type.

4 Conclusions

Additional instruments have been modeled using PhISM techniques. Pitched drums such as timbale and bongo have been explored using the PhISAM algorithm, and the PhISEM approach has been used to model and synthesize the tambourine and guiro. There still remain many instruments in the percussion family to be modeled, and the use of physical modeling coupled with parametric synthesis promises to provide expressive yet efficient sound synthesis.

References

Chafe, C. 1995. "Adding Vortex Noise to Wind Instrument Physical Models," *ICMC*, Banff, pp. 57-60.

Cook, P. 1995, "A Hierarchical System for Controlling Synthesis by Physical Modeling," Proc. *ICMC*, Banff, pp. 108-109.

Doutaut V. & A. Chaigne 1993. "Time Domain Simulations of Xylophone Bars," Stockholm MAC, KTH, pp.574-579.

Garton, B. 1992. "Virtual Performance Modeling," Proc. ICMC, Montreal, pp. 219-222.

Gould, H. & J. Tobochnik, 1996. *An Introduction to Computer Simulation Methods*, N.Y., Addison Wesley

Jànosy, Z., Karjalainen, M. & V. Välimäki 1994, "Intelligent Synthesis Control with Applications to a Physical Model of the Acoustic Guitar," ICMC, Aarhus, pp. 402-406.

Larouche, J. & J. Meillier 1994. "Multichannel Excitation/ Filter Modeling of Percussive Sounds with Application to the Piano," *IEEE Trans. Speech and Audio*, pp. 329-344.

Moore, J. 1970. *Acoustics of Bar Percussion Instruments*, Ph.D. Dissertation, Music Dept. of the Ohio State Univ.

Rossing, T. 1976. "Acoustics of Percussion Instruments - Part 1," *The Physics Teacher*, 14, pp. 546-556.

Serra, X. 1986. "A Computer Model for Bar Percussion Instruments," Proc. *ICMC*, The Hague, pp. 257-262.

Serra, X. & J. Smith 1990. "Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic plus Stochastic Decomposition." *Computer Music Journal* 14 (4), pp. 12-24.

Verge, M. 1995. *Aeroacoustics of Confined Jets, with Applications to the Physics of Recorder-Like Instruments*. Thesis, Technical University of Eindhoven, also available from IRCAM.

Wawrzynek, J. 1989. "VLSI Models for Sound Synthesis," in *Current Directions in Computer Music Research*, M. Mathews and J. Pierce Eds., Cambridge, MIT Press.