PHYSICAL MODELING FOR PURE DATA (PMPD) AND REAL TIME INTERACTION WITH AN AUDIO SYNTHESIS

Cyrille Henry
cyrille.henry@la-kitchen.fr

ABSTRACT

This article describes an application of physical modeling tools for pure data (pd) for real time interaction between a musician and an audio synthesis. pmpd is a collection of objects for pd providing a very flexible way to particle physical modeling simulation and other kind of comportment-based modelling. pmpd is used for data generation. That is to say that, although they can easily be used to *control* audio engines, they do not generate audio signals directly. Virtual physical structure can act as a black box between the musician and the audio synthesis. Using pmpd within the pd programming environment allows real-time interactions with this simulation, as well as natural control of a sound. A musician can play with the movement of a virtual structure, which produces sound.

1. INTRODUCTION

The system is mainly composed of software environment, mostly based on pd, GEM and pmpd. pd is a real-time generic programming environment dedicated to real time audio synthesis [1]. GEM adds Open-GL based 3D Graphics to pd. A visualization of the simulation can be created thought GEM, as in pmpd examples, but other video software could also be used. pmpd is a collection of "objects" for pd which make physical modeling possible. pmpd is an approach to models made of particles, which are only one of many options for dynamics systems. This approach is widely used for video animation and interactive simulation [2]. It provides a very flexible way to particle physical modeling simulation and other kinds of comportment-based modeling. Using pmpd allows real-time interactions with this simulation as pd is a powerful programming language in which all kinds of interaction with the user can be made. Moreover, pd programing environment can customize or enhance pmpd possibilities.

Pmpd is one of the many software dedicated to physical modeling [3][4][5]. The main difference is that pmpd is included in a very efficient and powerful software environment dedicated to audio synthesis. Pmpd can then very easily be used to control any kind of audio synthesis.

2. PMPD DESCRIPTION

pmpd is designed to provide low level comportment objects allowing particle-base physical modeling. Assembling these objects can generate complex behavior due to the interaction among the basic objects. A knowledge of the global equation of the movement is not necessary to simulate very complex behaviors. The cause of the movement and the structure only are needed for the simulation. pmpd can then easily be used for the simulation of a very large variety of comportments.

All the pmpd objects work with control data (as opposed to audio signals). For instance, one cannot hear the sound of a vibrating string because it will not move fast enough; but one can use the movement of these particles along a string to perform additive synthesis.

Complex simulations are basically made from two kinds of elementary objects: "mass" and "link".

"Mass" objects send position and receive force from "link" objects. "Link" objects receive the position of two masses and output forces for both of them.

pmpd does not use specific units.

2.1. Masses

"Mass" objects react like a point mass. "Mass" objects have inertia, but they have no volume (they cannot rotate). They take forces at their input, and output their positions.

For each time increment, position of a mass changes accordingly to the mass velocity, while velocity depends on its acceleration. The value of the acceleration is given by Newtonian dynamics:

$$\sum \vec{F} = m\vec{\gamma} \qquad (1)$$

When told, masses make the sum of the forces applied to them in order to compute their acceleration, and then deduce their new position.

2.2. Link

"Link" objects take two mass positions and output two opposite forces depending on the relative position and speed of the masses. Links are visco-elastic connections between two masses. The force generated by a link is :

$$\vec{F} = K\,\vec{X} + D\,\vec{V} \qquad (2)$$

where “K” is the rigidity, “D” is the dampening, “X” is the elongation of the link, and “V” is the relative velocity of two masses.

2.3. Forces and displacement

To allow real-time interaction, “mass” objects accept force and displacement messages from the user. pd can then be used to provide interaction between the user and the simulation. The extreme modularity of pd can offer a very large variety of interaction with the simulation thanks to sensors, haptic transducers, or classical computers inputs (mouse, keyboard). pd gives pmpd users lot's of interaction possibility.

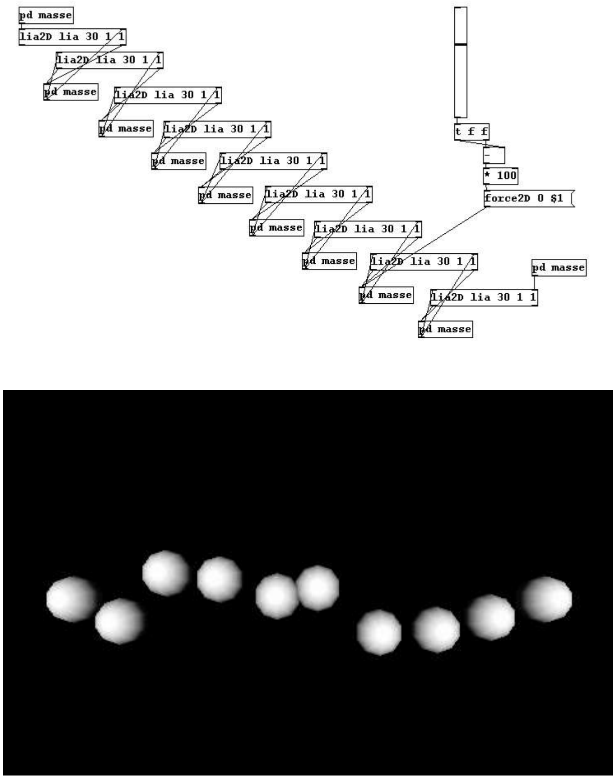

Figure 1. String example

The figure 1 shows an example of a pd patch modelling a simple string made with 2D masses, and a graphical representation of this vibrating string.

3. APPLICATION FOR REAL TIME INTERACTION WITH AN AUDIO SYNTHESIS

One of the applications of this kind of simulation is to create a dynamic structure that can be modified, moved, distorted by the user. This structure can then be used to “control” an audio synthesis. With the help of sensors, a user can create a virtual structure, linked to his or her own movement, to the real. The user can then play with a virtual, but “physical” instrument, allowing a natural comportment of a digital audio synthesis.

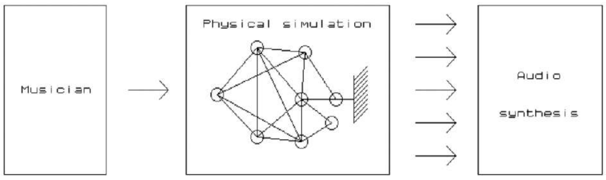

Figure 2. using physical modelling between musician and audio synthesis

The figure 2 shows a virtual structure used as a black box between user action and audio synthesis. This black box has interesting specifications. Only few input parameters can generate lots of different data flows (a musician can play with only few control parameters on the whole structure, and then generate lots of data to control any kind of audio synthesis). Moreover, the control parameters are intuitive because they correspond to physical values. Playing with such a system can be very intuitive for the performer as the system reacts in a instinctive way. Some control parameters can change the way the structure evolves within a time period. Another important specification is that all data coming out of the physical model are not independent. The relation between them can be adjusted regarding the topology of the structure.

One application of physical modeling structure used to drive an audio synthesis is known as “Scanned Synthesis” [6]. The use of the deformation of a string as an audio waveform was already explored [7], however pmpd offer a more flexible solution, easier than writing externals. Scanned synthesis is just an example of pmpd possibility, the end of this article describe 2 other very basic examples.

3.1. Bouncing ball

Figure 3 represent a “bouncing ball” structure. Forces applied to each masses control the amplitude of sinusoid which performs additive synthesis. The sound produced by the structure can be controlled while moving the structure, making it bounce, etc. The temporal evolution of the sound can be change with few parameters describing the global behavior of the structure such as rigidity or dampening of link [8].

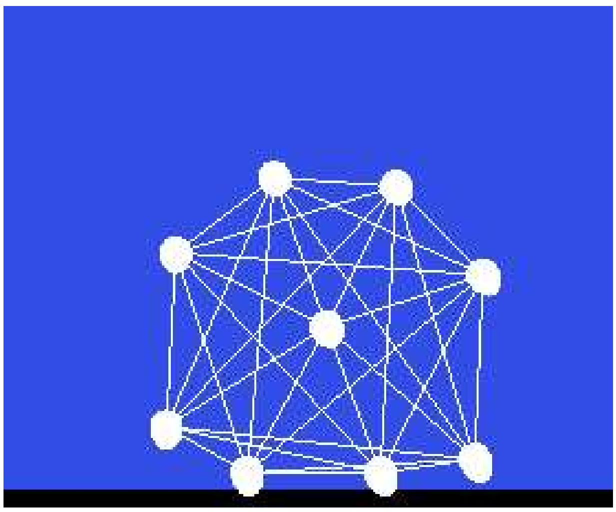

Figure 3. Bouncing ball used for additive synthesis

3.2. Rhythmic pattern

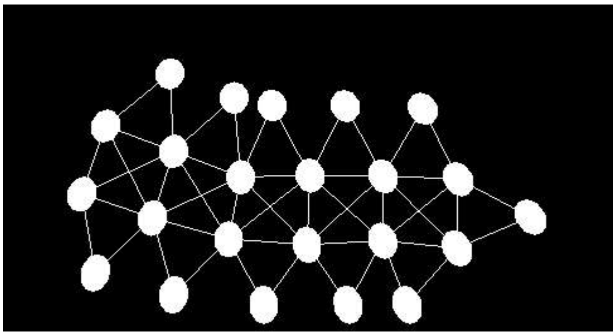

Figure 4. example of structure

The figure 4 shows a self-moving structure that can be used to trigger sound. Whenever a mass hits the floor, a sound is produced.

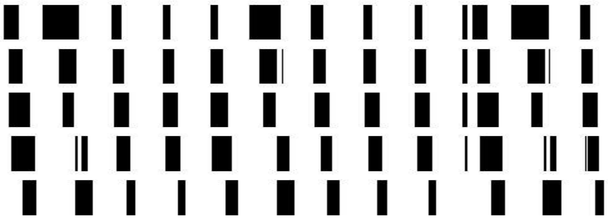

Figure 5. Rhythm generated by this structure

The figure 5 is a time representation of the generated rhythm. The main rhythm corresponds to the structure moving by itself while the internal deformations of the structure create the small rhythm variations.

4. PERSPECTIVES

The most commonly encountered problem while using physical modeling is the instability. To reduce the risk of instability, one's model should be slowed down (increasing the metronome speed can be necessary to keep the desired speed of the simulation). This is not a problem due to the relatively low frequency needed for the simulation. In this case, the structure needn't be computed at audio rate unlike physical modeling based audio synthesis, but at only a few hundreds hertz. So, this simulation can be calculated by low-cost computer or personal laptop for live application.

pmpd is released as free software under the GPL, and has been compiled on most common platforms (Linux, Windows, and Mac osX). Binaries, sources, examples and documentation can be downloaded from: http://drpichon.free.fr/pmpd

pmpd was also compiled for Max/MSP by Ali Momeni [9].

pmpd can be used for audio or video movement simulation. It can also be used for non real time audio data generation. Extension for real time audio synthesis is under development. However, limitations will come from the CPU speed of the computer.

5. REFERENCES

[1] Puckette, M. S. "Pure Data: another integrated computer music environment" *Proceedings of the International Computer Music Conference*, 37-41. 1996

[2] Castagne, N. Cadoz, C. "L'environnement GENESIS : créer avec les modèles physiques masse-interaction.", *Journées d'Informatique Musicale*, 9e édition, Marseille, 29 - 31 mai 2002

[3] S.Rimell,S D,M.Howard, A,D.Hunt, P,R.Kirk and A,M.Tyrrell *"The Development of a Computer-based, Physically Modelled Musical Instrument With Haptic Feedback, for the Performance and Composition of Electroacoustic Music."* Proceedings of the 10th anniversary European Society for the Cognitive Sciences of Music Conference, Liege 2002.

[4] Cadoz, C., A. Luciani and J.-L. Florens, "CORDIS-ANIMA : a modelling and Simulation System for Sound and Image Synthesis - The General Formalism", *Computer Music Journal* 17(1) 1993.

[5] http://web.ukonline.co.uk/taosynth/

[6] B.Verplank, M. Mathews, R. Shaw, "*Scanned Synthesis",* Proceedings of the 2000 International Computer Music Conference, ICMA, 2000

[7] Couturier J.M., "A Scanned Synthesis virtual Instrument", "*Proceedings of the 2002 Conference on New Instruments for Musical Expression*" (NIME-02), Dublin, Ireland, May 24-26, 2002

[8] Djoarian, P. "Material design in physical modelling sound synthesis" Proc. of the 2nd COST G-6 Workshop on *Digital Audio Effects* DAFx99, NTNU, Trondheim, Dec. 9-11, 1999

[9] http://www.cnmat.berkeley.edu/~ali/share/max/pmpd