Mass-Spring-System model for real time expressive behaviour synthesis

Why and how to use Physical Model in Pure Data (pmpd)

Cyrille Henry (ch@chnry.net)

Abstract

Mass-spring-system (MSS) physical model (PM) are vastly documented in literature and on internet. They where developed to imitate natural behaviours, specially for audio synthesis but they can also be used to create natural behavious or even "plausible" behaviours. This "plausible" behaviours are comportment that does not imitate any specific real life behaviours, but - since basic real life physic are respected - looks "natural". This is specially useful for real time user interactions. Even if this "plausible" behaviour can be use for any human / computer interface this document will mostly focuses on artistic application.

After a quick overview of the equation used for this kind of simulation, this document will list different strategies use in MSS (implementation, creation of a simulated system, strategies for interaction and using data from this system). Finally the pmpd library for pure data will be presented.

Introduction

In this document, we discuss the use of mass-spring-system physical model to create expressive movement for real time interaction.

We are dealing with the production and the control of data flow aiming at the control of any kind of software parameter : Sound synthesis, images synthesis, lighting or motor actuator by example can be driven thanks to a physical model. This parameters can usually benefit from an expressive control. We are using physical model to synthesize data flow and control their behaviours.

MSS model can be use to imitate natural behaviours, but they can also be used to create lot's of different behaviours. We did not focuses on real behaviours, but on "plausible" behaviours : a movement that look natural, even if it did not imitate a specific real life physics. MSS model can easily generate this "plausible" behaviours since it respect basic real life physics. Moreover this kind of physical models allow a wild range of user interaction for real time control of the movement.

Using this models can lead to control of different media with data flow and to control correlation between this data, thus having sound and images by example that does not represent the same data, but express the same underling process.

This document aim is to give key to a deep understanding of process involved in MSS model. After a quick overview of the equation used for this kind of simulation, we will present different approach to create topologies of mass / link network. Then it will list different strategies used to interact with this MSS simulation. Latter, we will list few approach to use the generated data flow for both sound synthesis control and images synthesis control. It should not be seen as exhaustive, but as a summarized of the author experience in developing PM for many different artistic applications. Latter, we will have a look at pmpd : a specific implementation of MSS system for the Pure Data software.

Index

Motivations

Real motion

Since a movement is initiated thanks to a force, looking at the speed and acceleration (that are linked to force, as we will see latter), is a good way to analyses a movement, and gives more details than only looking only at the position over time. Figure 1 represent position, velocity and acceleration over time for 6 different recording of a hand motion moving a cursor from 2 different points 10 cm approximatively apart.

Figure 1: 6 recording of a simple hand movement, and there corresponding velocity and acceleration over time.

Even for simple movement with relatively similar position vs. time curve, lot's of variety can appears in real life motion. When looking at the acceleration curve : minimum and maximum acceleration and position of this extreme can be relatively different. The human brain have been trained to analyse and understand this differences. Being able to reproduce such diversity is the first step toward generating expressive behaviours.

Movement using a simple interpolation

Using Pure data to move a parameter is usually done with the [line] object. This object perform a linear interpolation between two points (see figure 2). This is not a kind of movement that can easily be find in the nature. Using this to move a fader result in a very different movement than what would be performed by a human, mainly because it did not respect the world physics. The linear interpolation imply a constant speed of the parameter, and no velocity before and after the transition. This result in an infinite acceleration at the beginning and at the end of the movement.
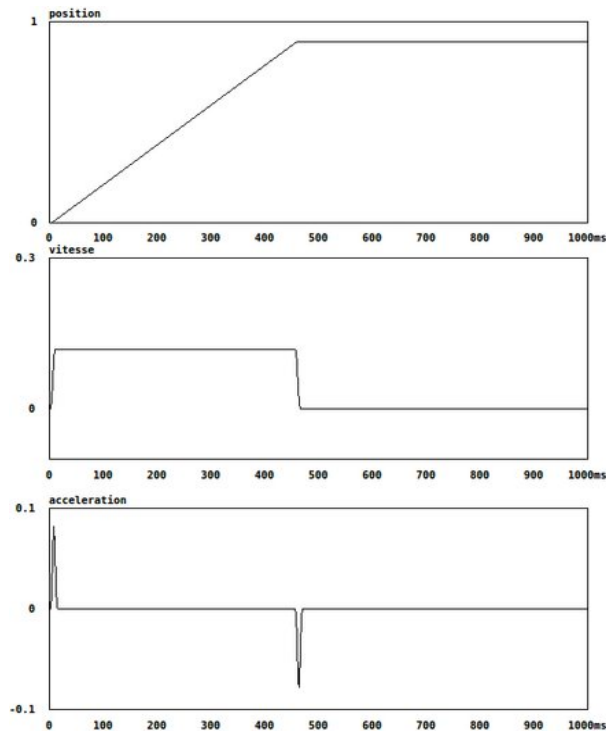

figure 2: movement generated by a linear interpolation

Such movement is easily identifiable as artificial.

Mass Spring System

Fdili Alaoui and Al explain motivation using a MSS in an article published in the IJPADM revue (REF 1). Here is an extract :

MSS present various properties that make them attractive to use in the performing arts (Hsieh 2007; Jacquemin 2008; Johnston 2009; Fdili Alaoui 2012). They have the ability to produce physically plausible behaviours because they are governed by the laws of physics and hence have an evocative representation of real-world dynamics. We describe the behaviour of a physical model as physically plausible when it generates realistically perceived dynamics that are created autonomously to evoke real-world dynamics, without imitating real-world mechanics.

MSS provide a wide range of controllable parameters in real-time (such as the parameters of the forces applied to the masses) that govern its behaviours, and thus offer rich possibilities of interaction. The MSS can be controlled through the mapping between the MSS physical parameters and the features of the analysed movement. The resulting interaction would create a relationship between the performer's movement and the associated MSS behaviours. This relationship follows pre-defined rules or interaction scenarios.

MSS simulations generate data streams that can control different media. The applications described in this article are limited to sound and visual media, but other actuators such as lighting devices can also be controlled. The data generated can be directly used as a synthesis kernel. For example the 'scanned synthesis' method by Verplanck et al. (2000) uses MSS to generate shapes, which build waveforms for audio synthesis. The use of MSS for a parallel control of different media can ensure a strong recognisable coherence between these media. The same physical simulation can generate data concurrently synthesized into sound and graphics. In real life a moving object's velocity and acceleration produce sound while the viewer perceives the object's position in time. Thus to generate plausible cross-media behaviours, one can simulate real life cross-modal phenomenon and use the mass positions to control the graphical rendering and their velocities and accelerations to control the sound synthesis. The parallel multimedia control allows us to perceive the audio-visual phenomenon with more consistency as both sound and visual behaviours come from the same physical simulation.
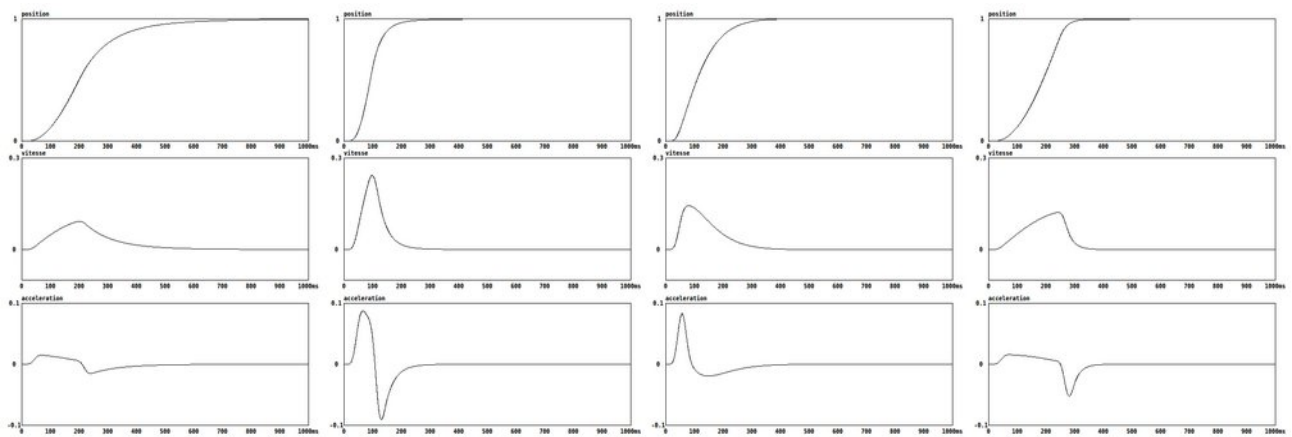

figure 3: different movement generated by a MSS

Figure 3 represent few movement that was created using simple MSS. This movements present few particularities that can be found in real life motion presented earlier.

Physical Model

Basic equation

MSS simulation are based on simple idealized physical elements :

- The mass is a punctual mass : it got weight, so it is subject to inertia, but it does not have size, so it is not subject to rotation. The mass react to newton equation :

$$\sum F = m\ \gamma$$

The mass is the moving part of the simulation. It react to input forces and output it's position. It's weight is the only adjustable parameter.

- A spring create an elastic connection between two masses. The spring create a force inverse to it's elongation or compression.

$$F = K * ((x_2 - x_1) - l_0)$$

The amplitude of the created force is proportional to the spring rigidity (K). It can also have a specific size (l_0). The spring have no mass.

The spring react to two masses positions, and interact with them sending two opposite forces. It allow kinetic/potential energies conversion, thus kinetic energy transfer from a mass to another.

- Just like a spring, a damper connect two masses. The masses positions are used to compute forces thanks to the viscosity (D) parameter of the damper. The force is proportional to the elongation speed of the damper.

$$F = D * (v_2 - v_1)$$

The damper is the only way for the system to lose energy.

Both spring and damper can be aggregate in a single visco-elastic link.

This equation can be computed in a one dimensional space : i.e. masses can only be moved in 1 direction and link are also projected in this direction. But they can also be computed in a 2, or 3 dimensional space, and they can also be easily extrapolated to a higher dimensional space.
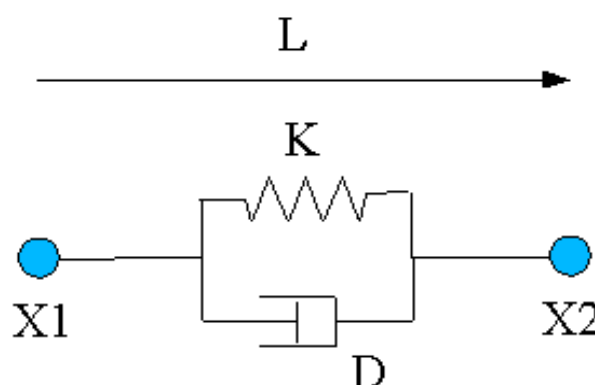

figure 4: representation of a spring and a damper connected as a single link

Implementation

In order to create a MSS simulation, one have to implement this equations on a computer. In the digital world, states of the model are computed in discreet time. i.e. displacement of a mass is not a continuous function, but a finite serial of point: x_n.

Position at time t is noted : x_t, while x_{t-1} is the position at previous time step.

The aim is to compute position of all masses at time t+1 depending on variables available at time t.

we usually compute velocity (v):

$$v_t = x_t-x_{t-1} / d_t$$

and acceleration (γ):

$$γ_t = v_t-v_{t-1} / d_t$$

In order to simplify the simulation, all constant can be removed : d_t (time difference between each computational step) is usually chose to be 1.

Since (for a mass) :

$$\sum F = m\ γ$$

so

$$v_{t+1} = v_t + \sum F/M$$

and

$$x_{t+1} = x_t + v_{t+1}$$

For the visco-elastic link (x1 and x2 are the position of the 2 masses connected by this link) :

$$F1_t = K(x2_t-x1_t) + D(v2_t-v1_t)$$

and

$$F2_t = -F1_t$$

One can noticed that changing weight of the masses in the same proportion as rigidity and damping of links did not change the simulation.

Some implementations uses only x_t and x_{t-1}, while other use x_t and v_t. The equation based on x_t and x_{t-1} are not provided but can be very easily deduced from the latter.

A PM simulation is based on a set of masses ans links. Links use masses position to compute the force they generate. Then, every masses add the forces they receive to compute a new velocity and a new position. This new position is send to the links in order to start again computing the forces...

Units

pmpd does not use specific units, you can chose your own. However, you should try to use units you that are consistent. i.e: you chose inches for the distance unit, then rigidity unit should be force unit divide by inches. For application describe in thius document, we just avoid using units.

Things are relative: a mass weighting 10 will react the same way to a force of 1 than a mass of 100 to a force of 10.

Basic oscillating system

Simple model do need only a very small processing power and implementation can be very simple. But they could be in great use in lot's of different applications.

One of the simplest examples of MSS consists of a single mass connected to a fixed point with a single link, forced to move along a single axis, and governed by an elastic and a viscous force. When the mass is displaced from its equilibrium position, its dynamics is controlled by the sum of an elastic force proportional to the displacement and a viscous force with a damping proportional to its velocity. When the mass is released, it oscillates with a fixed resonant frequency around the equilibrium position and with decreasing amplitude over time. Such a system is called a simple damped oscillator and the position of its mass over time is illustrated in figure 5.
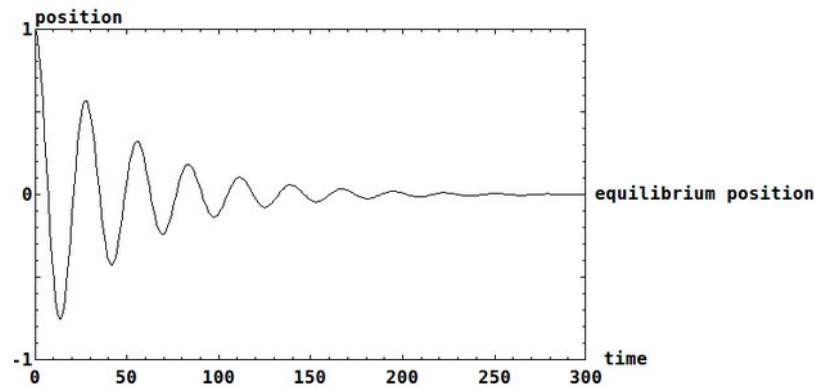

figure 5: The mass position of a simple damped oscillator after having been displaced from its equilibrium position.

Despite its simplicity, this system can be used in various ways. For example, when moving the fixed point, the mass position acts as a low pass resonant filter, while the spring elongation is a high pass filter.

When two simple damped oscillators are combined, the system's oscillations are more complex and result from the interaction of the two frequencies of each single oscillator. Thus, a MSS combining dozen, hundreds, or thousands of damped oscillators or more interconnected masses, can generate very complex behaviours that can then be used to metaphorically evoke human movements or to control any digital process. From a frequency point of view, the more masses composes the system, the more vibrating mode the system can admit : i.e. a 1 mass system can oscillate freely on 1 single resonant frequency, a 2 masses system can admit 2 frequency etc. Complexity usually arise with the masses number (or non linearity as we will discuss later).

Topologies

In order to use a MSS physical model, one have first to create a topology. i.e. a group of mass and connection between this masses. The topology can try to imitate a natural structure in order to mimic it's behaviours, but since we are also interested in abstract behaviours so we will focus on all kind of structure.

Regular structures

One of the easiest structure to experiment with are regular structure : string, nets or stack. The way masses are connect to each other is important for the global behaviours. For a 1d model by example, like the one presented on figure 6, the red connection are link between a mass and it's 2 neighbour and defined a string like system.
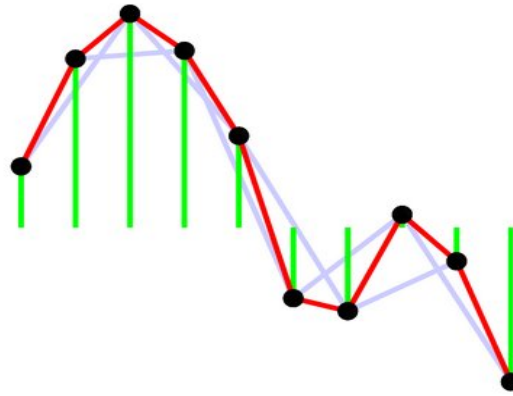
figure 6: topologies of a 1d string

The green connection (between a mass and a fixed central point) did not help energy propagation since they did not connect moving mass together. It can be useful to add damping, or allow faster up/down movement without changing the wave propagation.

The pink connection add flex rigidity to the structure.

Random connection can also be tested.

It is also important to manage borders : they can be "open" (when border masses are unconnected), in order to freely move. This will result on wave to bounce back with the same amplitude. They can also be "closed" : when border are fixed (masses connected to a fixed point), waves bounced back with opposite amplitude. It's also possible to create border that react in between this 2 opposite conditions. Using damped connection on the border it is also possible to remove any bounce back. One can also loop one side of the structure to the other, so that the waves does freely propagate. Of courses, different border condition can be used on the same model. Be aware that a model with 2 "open" border is not fixed and can then move away infinitely.

A frequent question regarding the 1d string come from the ascertainment that the masses can be represented next to each other on one direction, but moves on an other direction, perpendicular to the string. But since only the vertical displacement is used in the equation, all displacement of the masses are here only for visualisation of the structure. All masses are in fact on the same position for the simulation. Only the position in 1 direction is used. Hence the fact that this is only a 1d simulation.

Random structures

When creating this kind of structure, it's also possible to conceive them not really regular : masses weight, masses position, link length, rigidity or damping can be non uniformly created. From linear to random repartition, anything can be tested.
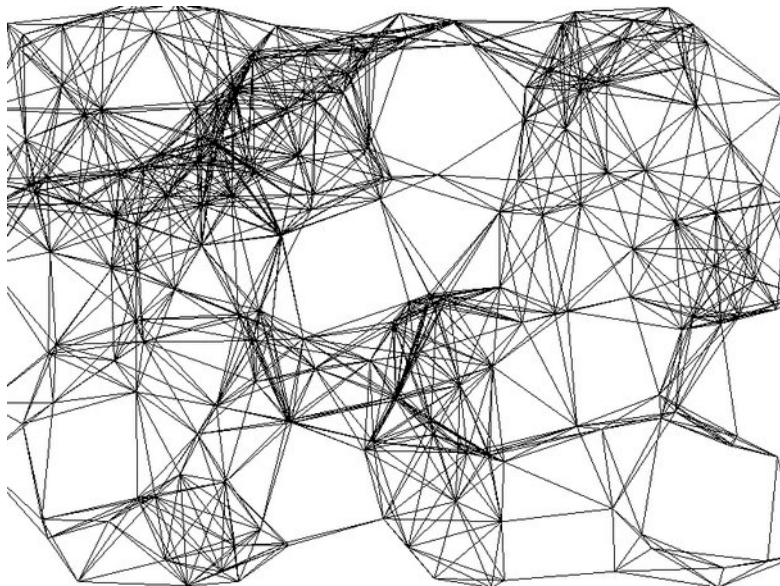
figure 7: structure using randomly positioned masses

Collection of independent simple structures

Complexity can also arise from the association of many very simple system. This simple system can all be identical, except initial condition, or physical property may differ a bit in order to introduce diversity in behaviours.

figure 8: few hundreds masses used for the egregore performance by chdh

Shape morphing

Simple shape morphing can also be archive using independent masses connected to different set of fixed point. Each set of fixed point defining the different global shapes. Adjusting rigidity of connection between all set of attracting point can result on masses being attracted by one of the shape. When adjusting rigidity of the various set of link, different Intermediate shape can be obtained.

figure 9: interpolation between a circle, a line and a point

In figure 9, 50 masses are connected to 3 different shapes : a circle, a line and a point. Each masses are connected with 3 links to this 3 shapes. This links are draw in different colour : red, green and blue for the connection with the point, circle and line. Adjusting the rigidity of the 3 sets of links change the shape created by the masses.
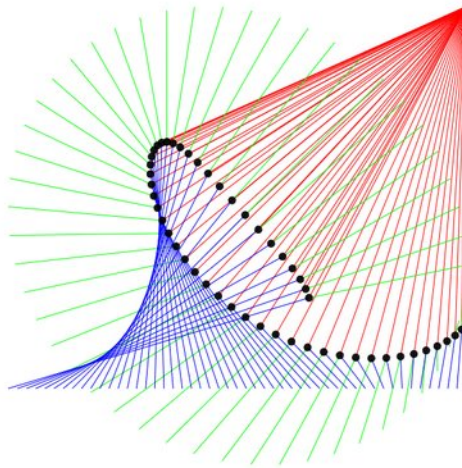
Modal synthesis

Simulation of a system can be decomposed in various vibration mode, each of them being modelled by a simple oscillator. Recombining this various oscillator result in a complex movement.

Even if dedicated optimisation can be made for modal synthesis, one can use MSS for modal synthesis. (REF 2)

Real time interaction strategies

In order to interact with the MSS, different approach can be experiments:

Moving fixed masses

Fixed masses are masses that are not moved by the simulation. But they can be moved by user input. When changing fixed masses position, the equilibrium position of the system is modified. The system usually oscillate around it's new equilibrium state, while damping tend to stop this oscillation.

Sending force to masses

Masses receive forces generated by links. But users can also explicitly send external forces in order increase or decrease the system energy. Forces can be send to one or to many masses. Forces can also depend to masses position, to create a simple or complex force field. By example, gravity can be simulate as a constant force field apply to every masses.
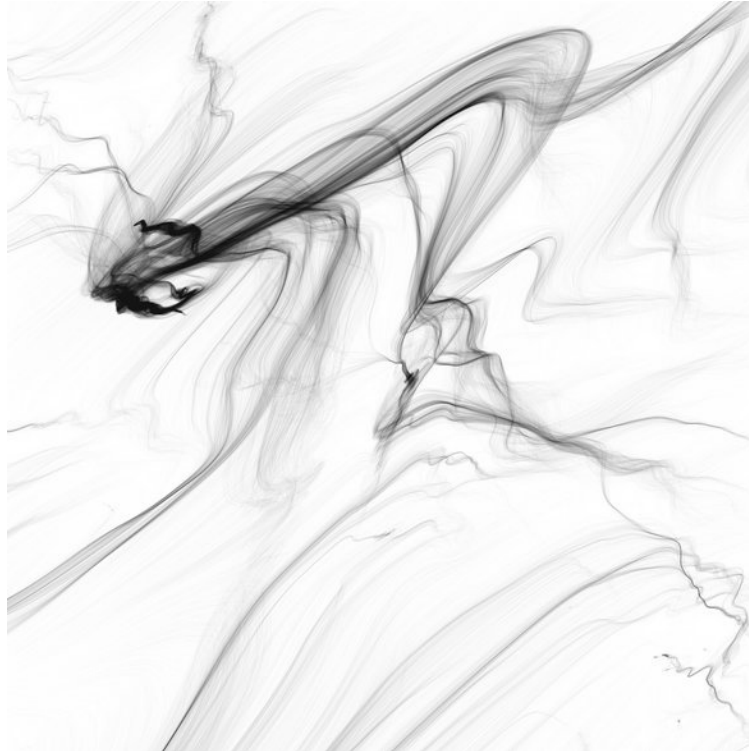
figure 10: "Traces" by Cyrille Henry

Figure 10 is an artwork made by traces left by many thousand identical masses in a complex force field. Each masses have a randomly distributed initial position. The force received by each mass depend only on it's position.

Changing physical parameter

Parameter defined during the creation of the PM topologies can also be changed during the simulation. Even if adjusting length of the links, rigidity and damping factor can lead to non realist simulation, it can still be a way to modifies the global behaviours. Physical parameters can also be set to non physical value. i.e. negative damping factor to increase energies, negative rigidity etc. However, great care have to be taken since this generally lead to a fast increase of the system energy that could result in system "explosion".

Using data created by a MSS

Using PM simulation is just a tool we are using to generate data. In our situation, data are generated continuously, in order to allow a real time interaction. This data can then control any other process. Sound, video or other physical actuator can all benefit form expressive control offered by a physical model. The way stage lighting, motor or other actuator can be controlled by a computer thanks to dedicate hardware and software is out of focus here. We will only discus the generation of data based upon PM simulation in order to control them. Latter, we will discuss sound or images synthesis based on MSS simulation.

We did not use the PM to mimic natural behaviours, but more like a black box (BB) allowing signal processing. Due to masses inertia, this BB is time dependent. i.e. output is not only a function of input, but also a function of time. Using this BB can also lead to simultaneous control of different media. For audio-visual synthesis, it allow sound and images not to represent the same data, while still being synchronize, so that both media emphasis the same underling process. (REF 3).

We already discuss the way users can interact with the simulation. All data coming out of

the simulation can be used. The most used are :

- Link length, elongation speed, and forces to be applied to masses : This data are computed for every link during the simulation process.

- Position and velocity are also computed for every masses. Acceleration is directly connected to input forces of the mass (as computed by links).

It is however possible to add link in the simulation that did not have any effect (using null rigidity and damping), in order to allow more data to be computed by the model.

Using large system result in a huge number of data. It is sometimes convenient to make some statistical analysis in order to reduce data quantities : average, standard deviation, local correlation of masses position, distance to the centre, masses velocity, forces can all be interesting depending the situation.

Sound synthesis

We just discuss many way to control a sound synthesis thanks a PM data. But it's also possible to create sounds entirely based on the PM.

Direct audio simulation

One can create a simulation of a vibrating object, "record" the position of a mass to create a sound file. Or use it in real time as an audio signal. When this document was initially written, processing power was a limitation to compute complex system at audio rate. But nowadays, convincing sound can be synthesise in RT with standard laptop.

Simple topologies like string, membrane, or even solid objects are the most obvious starting point to explore PM sound synthesis. But in 2d or 3d, even simple topologies can create complex sound. A double pendulum need only 2 links and 2 masses, but admit a chaotic behaviour.

Scanned synthesis

Scanned synthesis as describe in the literature (REF 4) consist of using *"a slowly vibrating object whose resonant frequencies are low enough so the performer can directly manipulate the object's vibrations by motions of his body and to scan (measure) the shape of the object along a periodic path by a periodic scanning function whose period is the fundamental frequency of the sound we wish to create."* Using MSS simulation, scanned synthesis usually consist of dumping the shape of a model on a table for each time iteration and reading this table at audio frequency.

Many improvement possibilities can be explore upon this original idea.

To increase sound quality, it is possible to read multiple time the same table in order to create chords of tones. It is also possible to read the table in one way, or back and forth, in order to change the spectral content of the produced sound.

Scan synthesis can also be seen as using a model shape as a wave shaper for a periodic sound input. The original definition can then be extended when using non periodic sound as input of the wave-shaper. Good result where obtained using filtered noise so that color and amplitude of the sound synthesis is altered by the model shape.

An other possible extension to this definition is not to use masses position to create a table that correspond to the shape of the model, but using other available data from the model such as masses velocity, acceleration or even link length. All of this result in different timbre and timbre evolution over time, offering a new range of experimentation for scanned synthesis.

Using scanned synthesis like a wave-shaper for a sawtooth signal usually result in sound which contain all harmonic of the fundamental frequency (with a limitation depending on the size of the wave shaper table). This harmonic series is similar in string instrument.

But one may desired to get only even harmonics or odd harmonics. Even harmonics can easily be obtained using a triangle sound wave as the waveshaper input. Having only odd harmonics like in tubes like instrument is a bit tricky. One solution is the use of a frequency shifter : when desiring a signal at frequency F with only odd harmonics, it's possible to create a signal at frequency 2F with only even harmonic (2F, 4F, 6F, 8F...), then shift it of -F in order to have frequency of : F, 3F, 5F, 7F etc. Using this trick can also be used to create other relation between harmonics, or non harmonic sound.

In some specific case, like using complex topologies or an important number of masses, interesting result have been obtained while ordering the scanned table. This result on a sound related to the global activity of the masses.

An other investigation field is the use of the shape of the PM model as a spectrum better than a waveform. This spectrum can be converted in sound using a reverse FFT.

"Many element synthesis"

A very different way to use a MSS simulation to create sound is to associate a sound synthesis for each simulated element (masses or links). Many experiment have been done using different synthesis strategies.

- additive synthesis : each masses controlling amplitude or frequency of an oscillator.

- subtractive synthesis : a noise or other complex wave-shape is filtered with many band cut filter.

- granular synthesis : each mass can be connected to a sound grain. The multiplicity of masses create a continuous and complex sound with evolution depending on the simulation.

All of this experiment have given interesting result depending on the system used and the sound desired.

Visual synthesis

MSS model representation can be quite simple : masses can be represented as points or spheres, and links as lines. It is also possible to add more information in the visualization : changing colour / size depending on force or velocity or any other simulation parameter. Lot's of other representation have been proposed in order to optimized volumetric representation.

On regular system like net, simple interpolation allow them to be seen as continuous. By example, masses can be used as control point of bi-cubic Bezier patch, in order to use them to distort a texture. By example, figure 11 is a net composed by 96x64 masses, that is rendered as a continuous system.
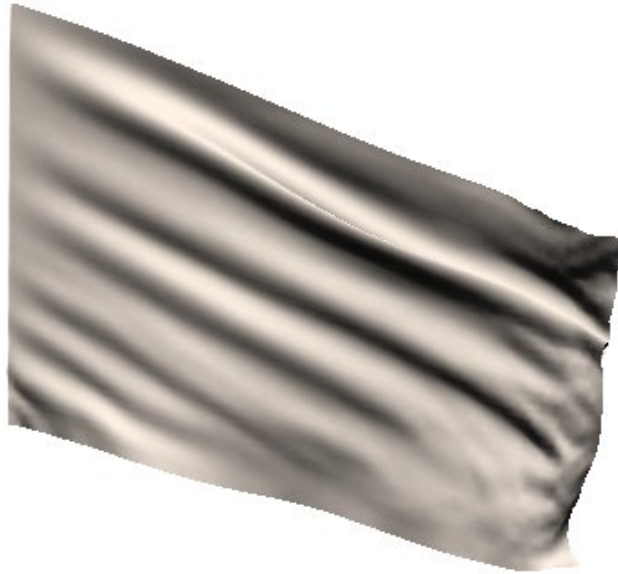
figure 11: simulation of a flag

But more abstract representation can be made when dissociating data from the PM and it's representation : by example, using forces to change points position, or position to change rotation of a shape, using X and Y Cartesian coordinate in a R Theta polar system etc.

Data analysis on the model can also be made, in order to represent only part of the simulated informations.
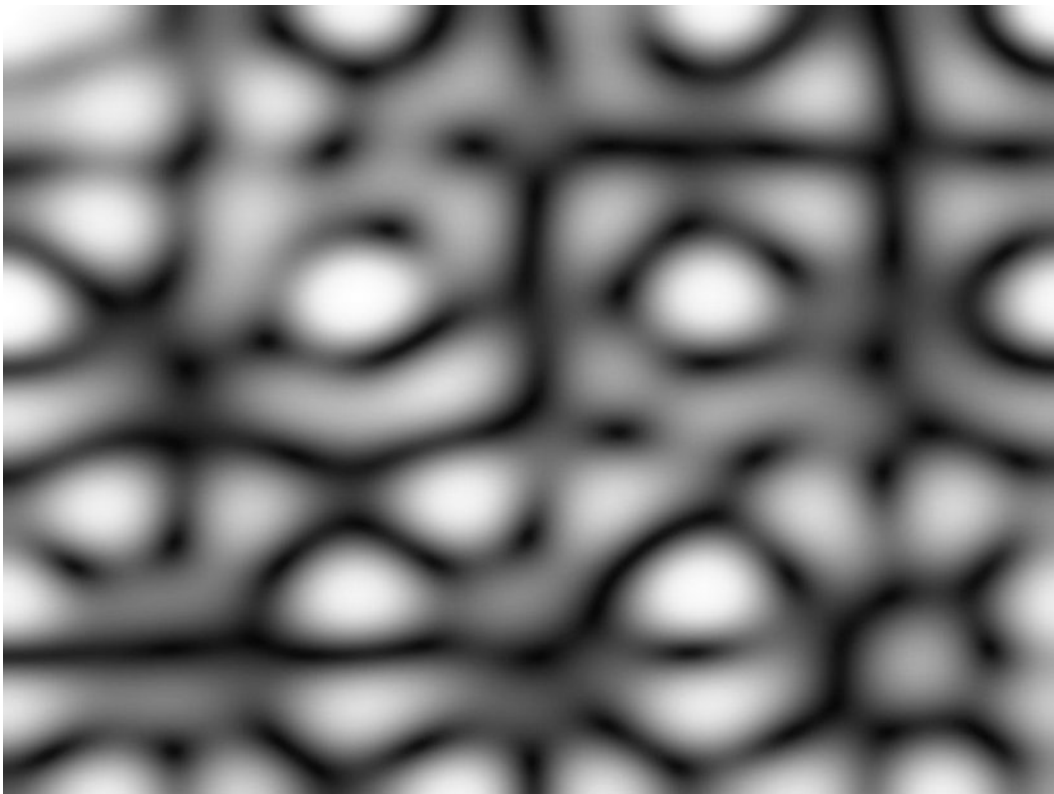

figure 12: amplitude visualisation of a vibrating membrane

By example, figure 12 is a representation inspired from Chladni plate : movement amplitude of a vibrating membrane is represented as colour intensity.

Complex system

Non linearity

Adding non linearity is a good way to add complexity in the simulation. There are many way physics can be non linear. Here are some easy way to add non linearity in the equation in order to respect -or not- physical laws.

Space limitation of links

Interaction between discrete object can be limited in space. By example, two objects can interact on each other only if they touch, just like a ball bouncing on the floor. This kind of interaction result in repulsion of the 2 objects, but not in attraction. It's easy to modify current equation to add a space limitation of forces generated by link. I.e. link do only create force if it's length is between a specific range.

Figure 13 is an example of using spatially limited link so that masses bounce on each other. This is used to create a simulation of bouncing ball inside the computer screen. (REF 5)

figure 13: bouncing ball "inside" the computer screen

Non linear rigidity

Some interaction in nature are not proportional to distance like in a spring. Gravitation force by example is inversely proportional to the square of the distance. It is possible to add an exponent (e) on the distance when computing the force create by a link in order allow different kind of interaction simulation :

$$F = K \, L^e$$

Using e = -2, a link simulate gravitation force. It is also possible to experiment other value for e :

e = 2 result in springs that rigidity increase with the square of the deformation. For a single mass-spring-damper oscillator, using this relation for the spring rigidity lead to oscillation those frequency depend on the amplitude. High amplitude result in higher frequency.

e = 0.5 result in a force proportional to the square root of the distance. Force generated by

high amplitude deformation result in weaker force than with a standard spring. Although this can help to prevent a model to diverge for high amplitude deformation, small deformations benefit of a very strong forces that can resulting on small instability. Small oscillation have higher frequency than bigger one.
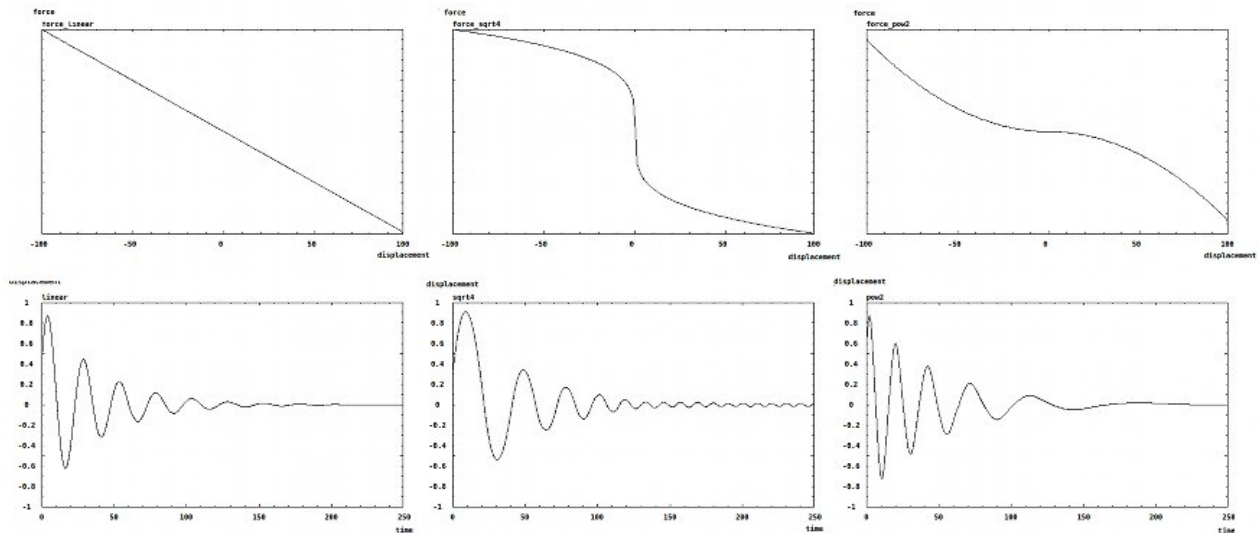

figure 14: non linear link and resulting movement of simple oscilator

Figure 14 represent the force vs. elongation of 3 different links, and the corresponding movement of a simple oscillator using this link.

Non linear damping

When using a simple function damping=D(v-v_0), no damping force are send when masses move at velocity v_0. But when masses goes faster, damping slow them down. When they move slower, the damping is negative and then generate a force that increase velocity. This relation for the damper is a way to impose a specific velocity (v_0) to masses.

Using a function like damping=D(v+v_0), forces always oppose to the movement so that masses can't stabilized and always oscillate.

On generic point of view, rigidity can be any function elongation, damping any function of velocity, and any external force (function of time and space) can be added. It's not mandatory in our situation to respect physics, but we will have a look at other possibilities to add complexity to equations in respect of physicals law.

More interactions

Motion damping

For an object moving in air, the air viscosity create a force that is in relation to the object velocity. This damping force can not easily be modelled by links, so it can be useful to add a velocity damping parameter to masses.

Creep of a link

Creep is the tendency of a solid material to move slowly or deform permanently under the influence of stresses. It occurs as a result of long term exposure to high levels of stress. It's possible to model an idealized creep in material : link length slowly change with time in

order to reduce solicitation.

$$L_{t+1} = L_t + r \, dX$$

r is link resilience. This allow the shape of the physical model to evolve with time.

Plastic link

Plastic deformation can also be modelled. (REF 6). It represent a system that permanently deform under a high solicitation. It can be simulated in the same way as the creep of a link, but only if the link force is greater than a threshold.

Breaking link

Modelling a link that break is quite easy : as soon as length of the link is greater than it's limit, rigidity (and damping) coefficient can be set to 0 so that it did not generate any force until the end of the simulation.
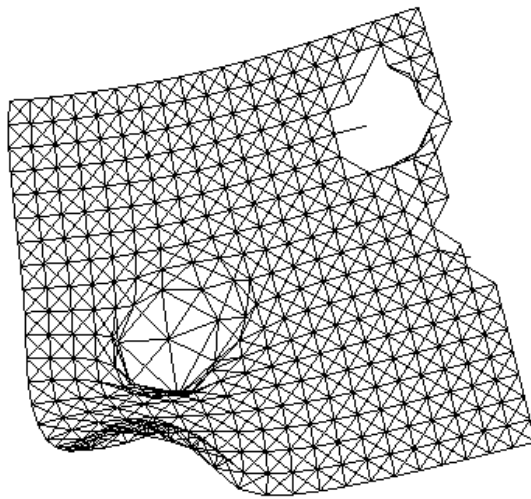

figure 15: pmpd example 64 : regular cube with distorted and broken links

Link elongation computation error

Let's consider a mass moving along the X axis, connected to the centre and admitting position x=-1 for time t-1 and x=1 for time t.

In a 1 dimensional space simulation, length of the link is computed by :

$$L = x2 - x1$$

So length can be negative. Velocity will be computed as :

$$F = D \, (L_t - L_{t-1})$$

In a 2 dimensional space simulation, length of the link is computed as :

$$L = sqrt((x2-x1)^2 + (y2-y1)^2)$$

For a mass being at position (-1,0) at time t-1 and position (1,0) a time t, the connection between this mass and point (0,0) will have a length of 1 for both time t-1 and time t. In this situation, damping will be null in a 2d computation, but not in a similar situation in a 1d simulation.

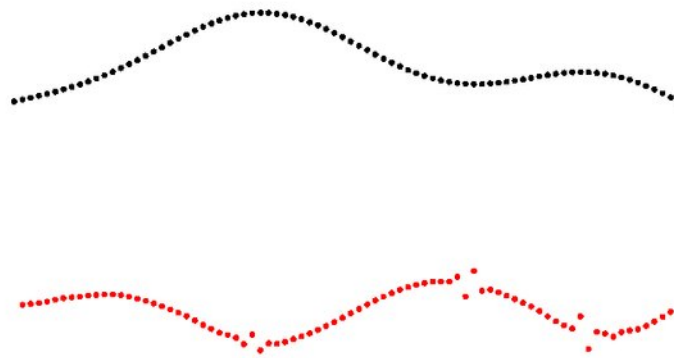This can generate simulation problems, specially in highly damped systems.

figure 16: error in a 2d string computation

Figure 16 represent a 2d string on top, while vertical forces are represented on the bottom. When the string admit a horizontal tangent, forces admit singularities that help to destabilized the system.

MSS simulation and emergent phenomena

Emergence is the way patterns arise out of a multitude of relatively simple interactions. Since MSS simulation involved a lot's of simple elements (masses and links), it's not surprising to look for topologies that would offer emerging spatial or temporal structures. Non linearity and an important number of masses are two important factor to create emergence. In the chdh "emergence" instrument that will be explore latter in this document, lot's of different spacial pattern have been created based on this concept. Statics patterns could arise, as well as dynamic one. Dynamic of this patterns goes from simple oscillations to non rhythmic behaviours.

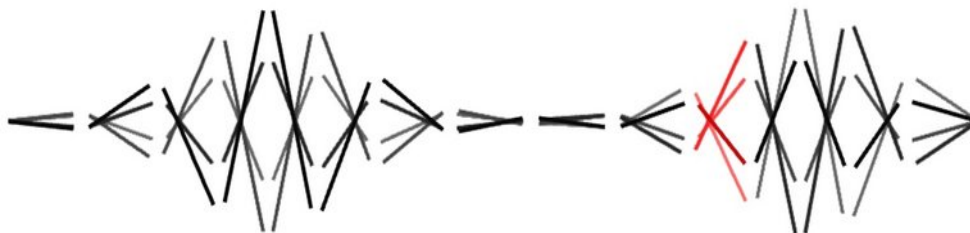But instability can also help emergence on simpler topologies.

figure 17: instabilities on a string

Figure 17 is a representation of a simple 1d string. Only links are represented with a line. Previous position are represented to offer a visualization of time evolution of the system. This is a 1d simulation, so masses can only move up/down. The right/left displacement is

only for clarity in representation.

Increasing forces can result in unstable pattern that can oscillate with time. See chdh vivarium album for more visualisation of this instabilities. (REF 7)

Emergence can also be part of the simulated model. Forcing one point of a 1d plate to oscillate create nodes that are the cause of chladni paterns (Figure 12).

Instabilities

When exploring physical model, the main problem comes from instability. An unstable system mean that the equation did not converge. This is usually due to too strong internal forces. Indeed, if rigidity or damping are too important, resulting on movement that would be too fast to be represented at a certain time increment, then equation are likely to diverge. (You can think of Nyquist to understand why movement can't be faster than the simulation speed!). When facing instability, the most obvious solution is to reduce forces involved in the simulation : rigidity, damping or external forces. The unwanted effect is that this slow everything down. Increasing system reactivity can be done by increasing the simulation clock speed.

Generally speaking, the simulation clock speed should be few time faster than the maximum frequency involved in the simulation. For interactive simulation, clock speed usually are around 50 to 100Hz, but some specific application require it to be increased to few thousand Hertz, or more. This is by example the case of the physical model used to model a flag in the wind in figure 11 : the apparent movement velocity is lot's slower than the cohesion force inside the material. In order to respect apparent movement, one have to correctly model this internal force, resulting in a high frequency clock.

Controlling instabilities

Numerical instability is usually an undesired effect in this kind of simulation. But it could however lead to surprising result, and thus be of some interest, specially if the simulation allow control of this instability. In "egregore", a chdh performance (REF 8), extensive use of "controlled" digital instabilities is made. Figure 18 is a snapshot of a chdh performance where instabilities is used to create various effects.
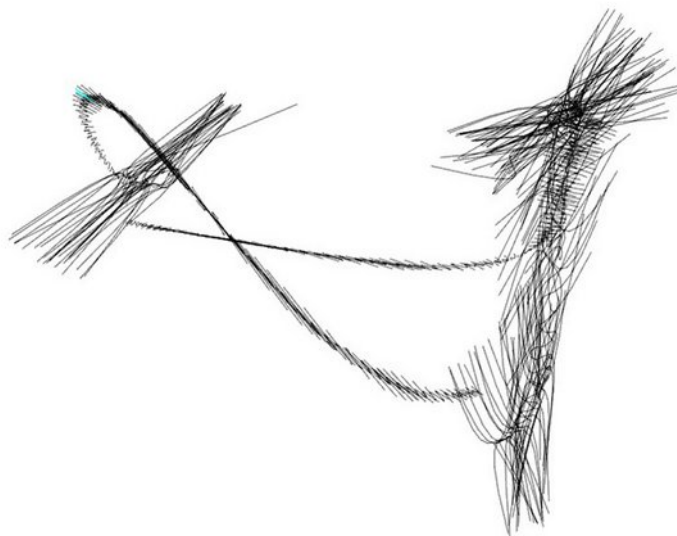

figure 18: instabilities from a shape morphing patch used in the egregore performance by chdh

Here are few tricks used to "control" instabilities:

- It usually is convenient to limit masses displacement to the minimum necessary for the simulation. This allow to limit the system divergence so it can converge back to a lower energy state when adjusting physical parameter correctly.

- Uniforms system instability usually propagate quickly in the whole system, resulting in a huge collapse, or explosion of the masses. Having instability that did not make the system to literally explode is quite challenging. The easiest solution is to use non linear spring: having a spring that rigidity is limited to a maximum value usually result in forces limitation, then in instabilities limitation.

- Non uniformity on a system can also create instability, that could however not propagate or diverge. Having by example a PM of a string using masses connected to each other. If one masses is lighter than the other, it's movement would be faster, so it will be the 1srt mass to became unstable when increasing spring rigidity. However, other masses could damp it's movement thus being able to control the instabilities amplitude of this masses.

- Due to geometries projection, some topologies can create non linear force even using only linear link. By example, let's consider a mass connected to 4 points disposed in a square, using link length the size of the square. Links are then compressed and repulse the masses. Anyhow, centre is a stable equilibrium for this system. Figure 19 represent the amplitude of the forces imposed by the 4 links to the mass, depending on it's position in space. Red point are the connecting point of the links.
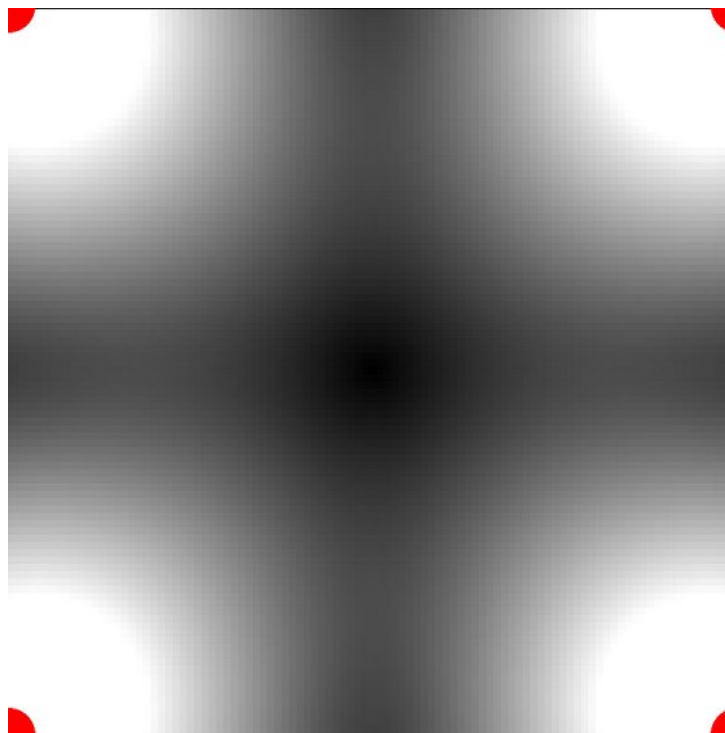

figure 19: Force amplitude depending of the position in space

Figure 20 represent the force regarding the mass position along the X axes. We can noticed that for small displacement, this 4 connections are equivalent to a null size spring connected to the centre. This is a stable equilibrium point, but increasing rigidity too much could create instabilities. However, the force decrease when displacement increase over a specific point, thus reducing forces. In this conditions, instabilities will not propagate.
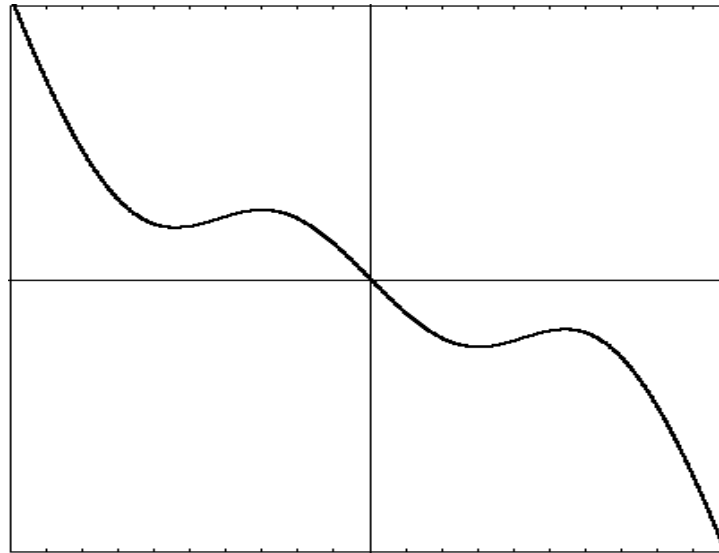
figure 20: force vs. X position

Example : chdh – emergence

Real life example as been describe in an IJPADM article (REF 1). Here is one of them :

Emergence is an instrument used to produce an audiovisual piece in the *vivarium* album and in the performance *egregore*. This instrument relies on a simple MSS, but produces complex behaviour. The model is based on approximately two hundred identical masses. Each mass is connected to all other masses with an identical and non-linear connection. Depending on the shape of the function curve that is used to compute the forces associated with each spring elongations, the masses organize themselves in different arrangements. While all the masses and links are identical, and no additional external force is applied to the masses, various spatial arrangement and rhythmical patterns can still emerge (Figure 22 and 23). It is worth noting that if the forces are too strong the masses will not stabilize in an equilibrium position, but will keep behaving in a spatially organized way, hence generating different kind of movement and visual arrangements.
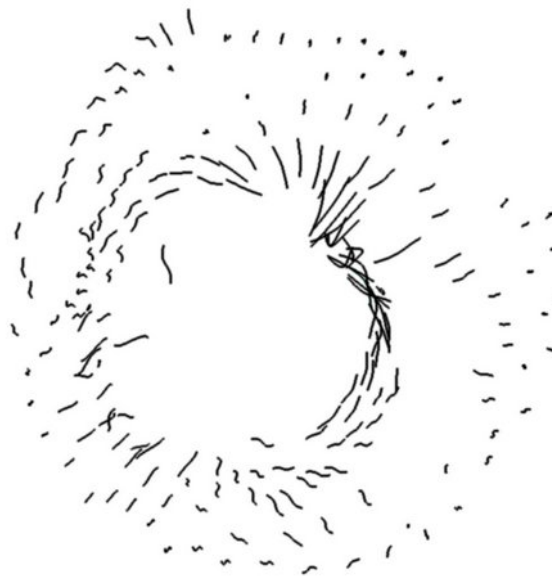

Figure 21: emergence of spatial arrangements in chdh visual instrument

Different function curves are used to associate the spring forces with their elongation. The control of all masses behaviours is made by interpolating between such predefined functions. Since the behaviours of the masses are very complex, we use a straightforward visual representation. Each mass is used to produce one sound and to visualize one curve described through the last mass positions. For each mass, we use an audio oscillator modulated by a filter. The velocity of the mass controls the filter frequency as well as the sound amplitude. The sound of the instrument is the addition of the sound produced by all masses.
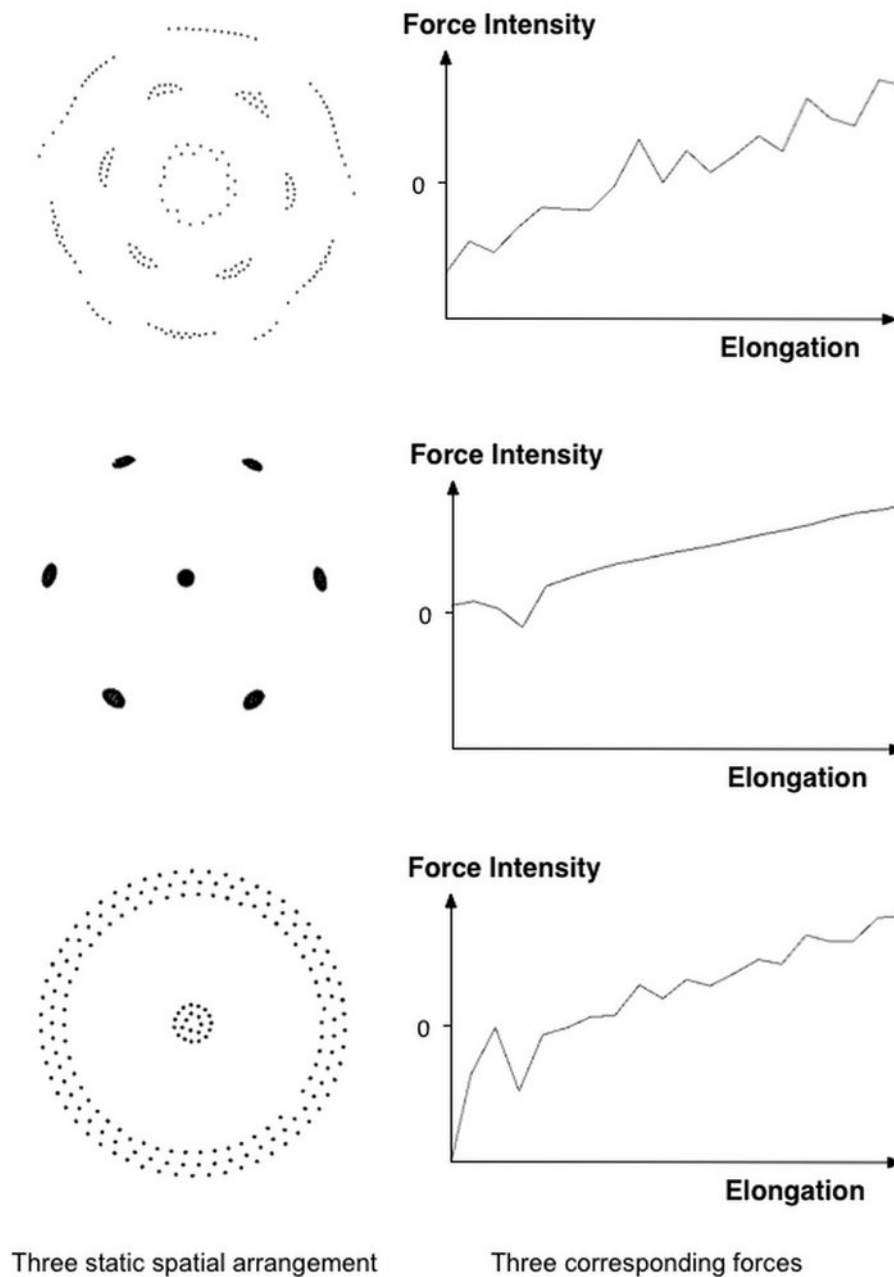
Three static spatial arrangement   Three corresponding forces

Figure 22: Different static spatial arrangements and the corresponding function curves used to compute the spring forces from their elongations.

Thanks to the use of a MSS, this instrument offers a very large range of complex behaviours, from organized structures to almost chaotic movements (Figure 23). Despite its multiple behaviours, the MSS can still be controlled in an accurate way because of the design choices used to select few available control parameters, allowing for its ease of use in real-time performance contexts.

In *emergence*, as well as in most chdh audiovisual instruments, sound and images are created using two different sets of data coming from the same MSS. In this work, the sound is never a direct representation of the images, nor the image of the sound. Nevertheless, they share a common source of control. Since both media represent the same abstract behaviours, the audience perceives a strong connection between them. Overall, the feedback of the audience about chdh performances is that instruments are described by the public using analogies of their movement or compared with animal behaviours or physical phenomena : it's the movement that define the instrument, not the sound nor the visual representation of the the movent.
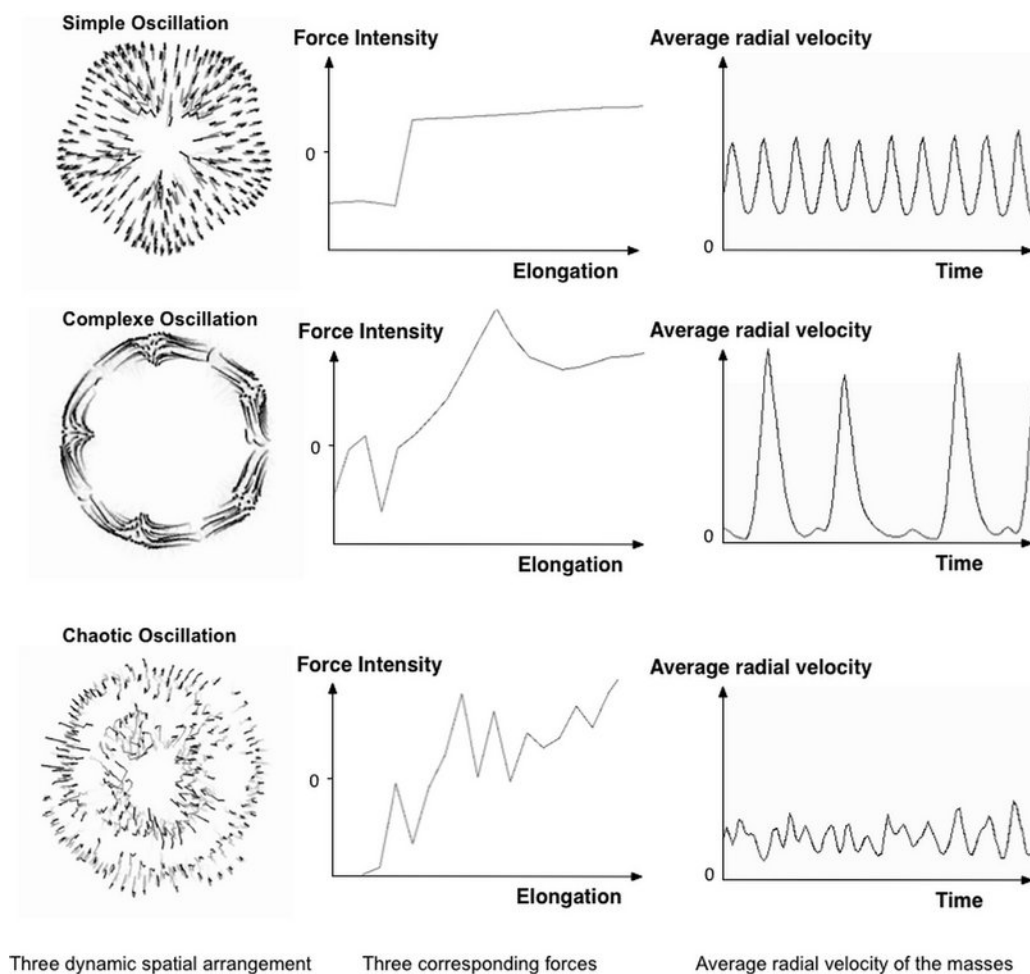

Figure 23: rhythmic patterns i.e. spatial mass configurations, function curves used to compute the spring force as a function of its elongation, and the mass average radial velocities over time in three different cases: simple oscillation, complex, and chaotic rhythms.

MSS and Pure Data: the pmpd library

Pmpd (physical modelling for Pure Data) is a library developed for Pure Data (Pd) (REF 9) aiming at physical modelling simulation. This library provide object that react like a mass or a link. Connecting this objects together allow the creation of any topologies. For performance reason, pmpd provide also all in one objects that can be controlled via messages. Pmpd have been developed for 1, 2 or 3d simulations.
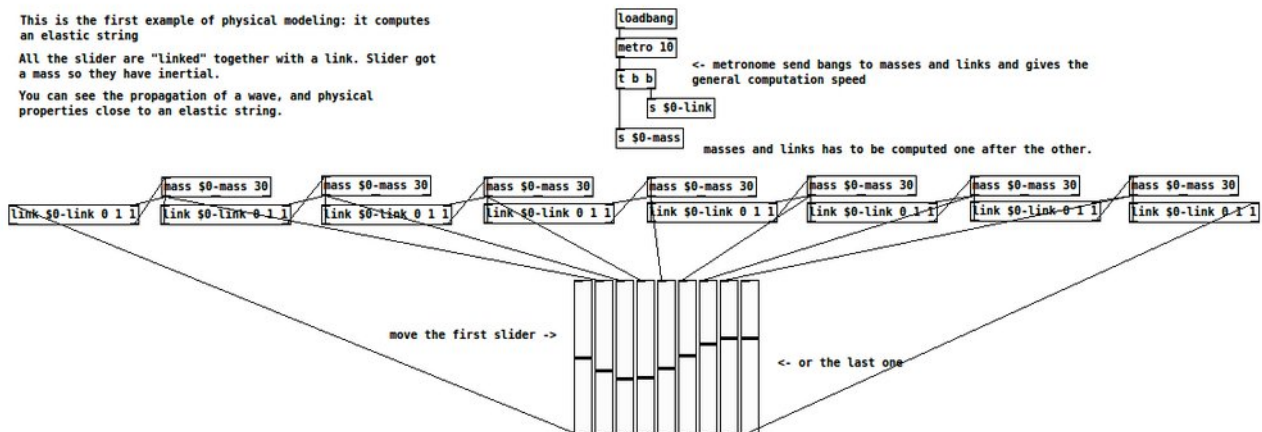

figure 24: string simulation in pmpd

Figure 24 present a simple string simulation using pmpd.

Generality

Most of the objects parameters can be set at the creation of the object as argument, but they can be modified using a messages sent to the first inlet.

For patching simplification mass and link objects have a name. It is the first argument for the object creation. It is used to receive information (pd messages). All masses with the same name defined a class of mass.

Pmpd did not check for physical validity of parameters. For example, you can set damping to a negative value, which mean energies creation. This is not physical and can lead to instability or saturation of the model.

Masses

Masses object admit a weight and an initial position. They receive force that change there velocities and output there current position.

"mass" objects also have outlets for their internal state; they report their position, the total force applied to them and their velocity.

Link

Links are visco-elastik link. They have an initial length, rigidity and damping value. They have 2 inlets for the position of there extremities, and compute 2 opposite forces depending on there elongation and elongation speed.

Metronome

A time reference is needed to compute the simulation. pmpd uses an "external" scheduler.

the user has to send a bang event to pmpd object.

This mechanism was chosen purposefully. The desired advantages were:

- easy change the speed of the simulation

- possibility to easily start and stop part of the simulation

- synchronisation to video rendering if desired

- split mass and link computation

The metronome corresponds with the time discretization of the equations.

pmpd does not use specific units.

Interactor

Interactor objects act as link object but provide patching facility.

In effect, a single object can create an interaction with a entire class of masses.

Interactor objects should be created with a name. This name is the name of the masses it is connected to.
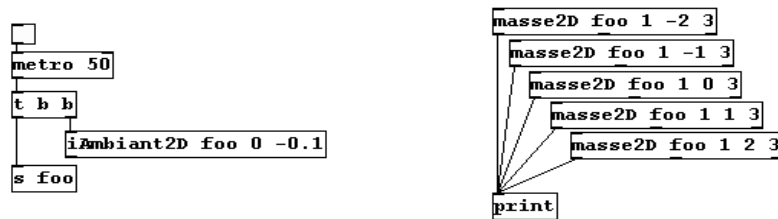

figure 25: interactor used to send a for to a class of mass

In Figure 25, all masses are subject to an ambient constant force. This force can viewed as gravity force applied to every mass named "foo".

Masses with other name will not be sensitive to this interactor. Different interactor objects provide interactions with a point, a line and other simple primitives.

Test

The objects test the position, as well as distance, speed from a point, a line, orientation, of a mass. Thereby, pd has access to much informations regarding the state of the system. This allows interaction with the rest of the patch. Another test object gives information about the link (deformation, speed of deformation, orientation...)

pmpd* objects

Pmpd, pmpd2d and pmpd3d are 3 almost similar objects, that allow simulation in 1, 2 or 3 dimensional space. Only 1 of this object is needed for a simulation. Masses and links are created "inside" the object. Creating, interacting and getting data from the physical model is allowed by sending and getting messages to the object inlet and outlet.

System creation

Messages "mass" and "link" followed by various arguments allow the creation of masses and links. Arguments for masses are :

- name of the mass that is use to identify it. Many masses can have the same name. They can also be identify thanks to a number relative to there creation order.

- a flag that allow the mass to act like a fixed point or not.

- a weight.

- an initial position (1, 2 or 3 number depending on the dimensional space the simulation take place)

links arguments are :

- name of the link

- 1st mass it is connected to.

- 2nd mass it is connected to. If a mass is designed by a name that is shared by many masses, then many links will be created with one single message.

- Rigidity constant

- Damping constant

(more arguments can be used to change default value of some other parameters)

Interacting with the system

Lot's of messages are available to interact with the system. Most used are setL, setK, setD to change links proprieties. This messages can interact with one specific link (thanks to it digital id), or to many of them in the same time thanks to the name provided during creation of the link.

One can also send forces to one or many masses thanks to the "force" messages.

Changing position of "fixed" masses is possible thanks to the "setPos" message.

In case many different value have to be send to change many masses or link properties, most messages allow the use of a table as source of this data.

"interactors" messages allow masses to interact with a force field. They are not persistent like a link : they act 1 time like forces messages.

See object's help file for more informations.

Getting data from a system

Few message ask the object to output information on masses position, velocities or link properties in various format : single element, list or directly on a table.

Some messages ask for statistical information about the simulation : average or standard deviation are available for various parameters like velocity, position center distance etc.

An other important message is the test message that allow to filter masses or link that passes different test on the same time. By example it's possible to get all masses named "foo" that velocity is greater than 3, and X position lower than 2. It is then possible to adapt the simulation thanks to this specific information.

All of the messages send and received by this objects are presented in each object help files.

Pmpd*~ objects

Pmpd1d~, pmpd2d~ and pmpd3d~ are similar to pmpd1d, pmpd2d and pmpd3d respectively, but they are simplified in order to optimise computation. They accept forces and position as audio stream, and also output audio signals.

Message are used to create masses, links, but also connection between inlets/outlets and the simulation. An audio inlet can control position of a static mass, or can send force to specific masses. Audio outlet can output position of a mass, forces it receive, or the sum of multiple informations.
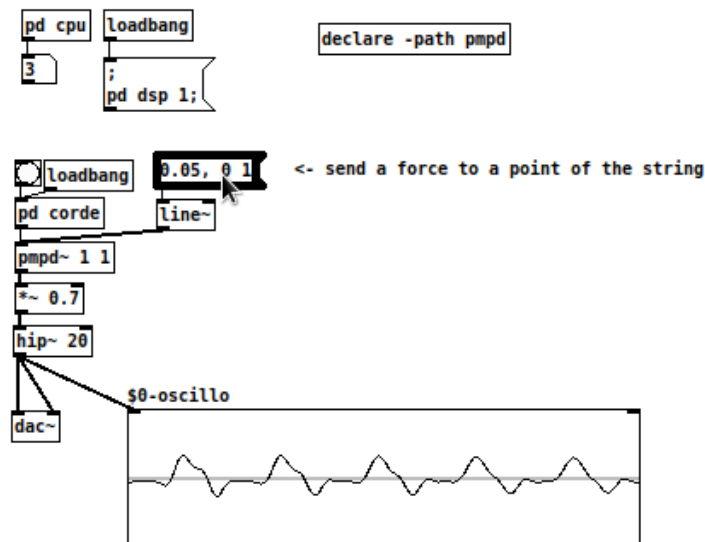

Figure 26: simulation of a string in pmpd1d~

website

http://www.chnry.net/ch/spip.php?article81

Sources

https://github.com/ch-nry/pd_pmpd

Installation

Pmpd can be installed like most Pd externals : using deken.

Conclusion

Physical Modelling is a powerful tool to generate data aiming at the control of any media. It can be use to generate a wide range of movement, and more specially realistic or plausible behaviours.

Pmpd is an implementation of such equations that is design to work on the Pure data programming environment. This object library was successfully used by many artist to help the creation of various art piece, from performing art, installation, or even picture or sculptures, since 2003.

Annexes

References

1 - Sarah Fdili Alaoui, Cyrille Henry & Christian Jacquemin (2014) Physical modelling for interactive installations and the performing arts, International Journal of Performance Arts and Digital Media, 10:2, 159-178, DOI: 10.1080/14794713.2014.946284

2 - J M Adrien , The missing link: Modal synthesis, Representations of Musical Signals, Pages 269-298 , MIT Press Cambridge, MA, USA ©1991

3 - Ali Momeni and Cyrille Henry, "Dynamic Independent Mapping Layers for Concurrent Control of Audio and Video Synthesis", Computer Music Journal, 30:1, pp. 49-66, Spring 2006

4 - Verplank, Bill; Mathews, Max; Shaw, Robert – ICMC, Volume 2000, 2000 - Scanned Synthesis

5 - http://www.chnry.net/ch/?075-Sables

6 - http://www.chnry.net/ch/?081-pmpd

7 - http://www.chdh.net/vivarium

8 - http://www.chdh.net/egregore

9 - http://msp.ucsd.edu/software.html

pmpd objects list

Mass : the 1d mass object

mass2D : the 2d mass object

mass3D : the 3d mass object

link : the 1d link

link2D : the 2d link

link3D : the 3d link

iAmbient2D : ambient interactor for 2d simulation

iAmbient3D : ambient interactor for 3d simulation

iCircle2D : circular interactor for 2d simulation

iCircle3D : circular interactor for 3d simulation

iCylinder3D : cylindrical interactor for 3d simulation

iLine2D : linear interactor for 2d interation

iPlane3D : plane interactor for 3d simulation

iSeg2D : segment interactor for 2d simulation

iSphere3D : spherical interactor fir 3d simulation

tCircle2D : circular test object for 2d simulation

tCircle3D : circular test object for 3d simulation

tCube3D : cubic test object for 3d simulation

tCylinder3D : cylindric test object for 3d simulation

tLine2D : linear test object for 2d simulation

tPlane3D : plane test object for 3d simulation

tSeg2D : segment test object for 2d simulation

tSphere3D : spherical test object for 3d simulation

tSquare2D : square test object for 2d simulation

tLink2D : provide information for 2d link object

tLink3D : provide information for 3d link object

pmpd : single object providing hight performance 1d simulation

pmpd2d : single object providing hight performance 2d simulation

pmpd3d : single object providing hight performance 2d simulation

pmpd~ : single object providing simulation at audio rate, 1d simulation

pmpd2d~ : single object providing simulation at audio rate, 2d simulation

pmpd3d~ : single object providing simulation at audio rate, 3d simulation