A Banded waveguide model of the Tibetan singing bowl

Mathias Kirkegaard
Final Project MUMT618 - Supervisor: Dr. Gary Scavone

Abstract

A model of the tibetan singing bowl has been succesfully implemented in matlab, using banded waveguide theory and a nonlinear interaction model for sustained excitation. Noise was added to the model, in order enhance the sound of dry friction.

Contents

1. Introduction

The Tibetan singing bowl is a percussion instrument with origin in Buddhist religious practices around 500 to 125 BCE. It was used to accompany meditation- or healing ceremonies, and later became an instrument of great interest in modern western music.
The following section will go through the key timbral characteristics of the instrument and thereby explain the objectives for this project.

Timbral characteristics

The Tibetan singing bowl has a rich timbre, constituted by a large number of inharmonic modes, i.e. non-integer frequency ratios k:

$$k = \frac{f_k}{f_0}, \quad k \notin \mathbb{Z} \tag{1}$$

Traditionally, the bowl is hand-hammered into the desired tuning, yielding a slightly asymmetric shape. This results in a strong beating in the lower modes, which is highly audible and a key timbral character of the instrument.
The bowl is played with a wooden mallet, a *puja* see section 1, by either striking or rubbing in a sliding motion along the rim of the bowl.
When struck, there is a distinct transient metallic sound, followed by a high number of inharmonic resonances decaying very slowly.
When rubbed, the energy buildup is slow and only a subset of modes are excited, depending on the velocity and pressure applied to the puja.
Furthermore there is a distinct sound of dry friction between the puja and the bowl.
Some work has been done on modelling this ancient instrument, most prominent is [7, 2] where a physically informed 1D banded waveguide approach is taken in order to obtain a real-time applicable model. This was followed up in [9], with an attempt to introduce more physical accuracy, mainly by taking into account both radial and tangential parameters, thus extending the model in 2D.

This project aims to model the bowl in Matlab with banded waveguides as proposed in[2], though using measurement results from [9].

There is an already existing implementation in the C++ library STK (synthesis toolkit) called `prayer bowl` in the src file `BandedWG.cpp` [P. Cook, G. Scavone], which will serve as inspiration and comparison.
The STK implementation lack the direct frictional sound of rubbing, and this project attempts to address this feature.
The objectives is thus a banded waveguide model that implements following features:

- Long decay
- Inharmomic modes
- Beating modes
- Sustained excitation
- Enhanced dry friction sound

2. Digital waveguides

This section summarizes digital waveguide theory including a thorough explanation of banded waveguides, which is used throughout the rest of the report.
Digital waveguides is widely used as a means to simulate wave-propagation in an acoustic instrument, for real time applications. One dimensional waveguides has been used for real time modelling of plucked[1] and bowed strings[5], open and closed acoustic pipes and reverberation. The main idea is to use bi directional delay lines connected in a loop, to represent traveling waves along a physical dimension of an instrument, and filters to handle reflections and losses at extremities. A digital waveguide model can thereby naturally represent the harmonic standing waves that appear on e.g. strings and pipes, some time after excitation.
A unit delay of $\frac{1}{f_s}$ translates into propagating the distance $\frac{c}{f_s}$ mm, where c is the wave propagation speed. For waves propagating in air this corresponds to a spatial sampling distance of 7.7 mm. At every time instance $\frac{n}{f_s}|n \in \mathbb{Z}$ pointers are incremented to simulate wave propagation, and a filtering operation is performed. Thus, digital waveguides is computationally inexpensive while maintaining a high level of physical accuracy.[5].

This theory have been extended to modelling in both 2 and 3 dimensions, as well as in a mesh structure, i.e. a "waveguide mesh" where the geometry of rooms or complex shaped resonators, is modeled by a grid of unit delay lines and multiport scattering junctions, [8, 4] an approach very similar to that of finite element modelling.

2.1. Banded waveguides

Banded waveguides is an extension to the 1 dimensional waveguide, and specifically a special case of digital waveguide networks. It was originally proposed in [2], as a means to model friction driven inharmonic resonators, such as the glass harmonica, bowed bars and the tibetan singing bowl.

The idea is to use parallel bandpass filters for decomposing the response of an instrument into sub-bands that represent its modes, i.e. significant peaks in the frequency response. The bandlimited signals are then fed into bi-directional delay lines of length $k\frac{f_s}{f_m}$, to model wave propagation of each mode separately. The choice of k will not affect the steady state response, but it will have an impact on the transient response, as it defines when a mode starts to "speak". This approach allows for modelling inharmonic resonances, as the centre frequency and delay length of each mode is tunable. An illustration is seen in fig. 1, notice how the channels are summed and coupled with an interaction block. For modelling the Tibetan bowl, this is the friction model eq. (4) that depends of incoming traveling wave components.

All losses are lumped together and implemented in the bandpass filters. This leads to 4 model parameters: modal frequencies, passband gains, beating ratios and delay lengths, which are retrieved in an analysis step that could involve spectral measurements and some geometrical considerations. This project will only focus on the synthesis part, as measurement results are taken from [9] and the STK implementation.

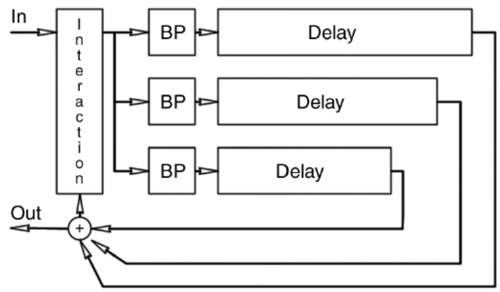


Figure 1: Simplified banded waveguide structure as proposed in [2]

3. Analysis

This section gives a quick overview on the analysis results that was used for deriving the model parameters.

3.1. Mode frequencies

In [9] measurements where performed on 4 bowls with diameters and fundamental frequencies given in table 1, to obtain 4 sets of frequency ratios. The matlab script `bowlParameters.m` implements preset selection based on matching the desired `f0` to one of the 4 measurements.

ϕ [mm]	f_0 [Hz]
262	86.7
180	219.6
152	310.2
140	513

Table 1: Bowl measurements from[9]

The actual mode frequencies is then calculated by multiplying the desired `f0` with the ratios in the selected preset.

3.2. Transient excitation

The excitation signal is normally a short noise burst retrieved by inverse filtering of a recording. In STK, the noise burst is implemented by simply pre-filling the delay lines with unique square waves of different pulse widths, and this method is adapted in this project. The procedure is illustrated in the code-snippet below.

```
minDelay = delays(end);
for k = 1:Nch
    for j = 1:(delays(k)/minDelay)
        x(k,j) = excitation(k)/Nch;
    end
end
```

`excitation` is a vector of size `(1,Nch)`, representing the amplitude of the square wave for each channel.
This method effectively implements a phase aligned transient attack, as the pulse widths is calculated from delay lengths, such that the rising edge for each channel is synchronized.

3.3. Beating ratios

The beating frequencies are given for the 3 small bowls[9], and they are in the range [1.001 1.021]. There seamed to be no correlation across mode numbers and bowl sizes, and therefore the parameter is left as user input that scales all beating modes equally.

3.4. Delay line length

The delay lengths are calculated from the modal frequencies as $D = k\frac{f_s}{f_m}$. With $k = 4$ for the 2 lowest modes, to make them "speak" later, effectively simulating a energy transfer from high to low frequencies, as is the case with cymbals.

4. Synthesis

This section will explain the steps taken in order to synthesize a sound, given the parameters in previous section. It is divided into two parts: the resonator and the sustained excitation.

4.1. Resonator

The resonator is synthesized with banded waveguides as described in [2], though a few changes were made in order to increase flexibility and to ensure stability. These changes are explained below.

Beating

A beating results from mixing 2 signals with slightly different frequencies. This is implemented by 2 detuned waveguides in series with 2 bandpass filters, illustrated in fig. 4. This differs from the circular banded waveguides proposed in [2] which used only one bandpass filter, see Figure 2. Separating the detuned waveguides allowed for changing the amplitude of the beating, by adjusting the passband gains differently, but also led to an issue of stability described in the following.

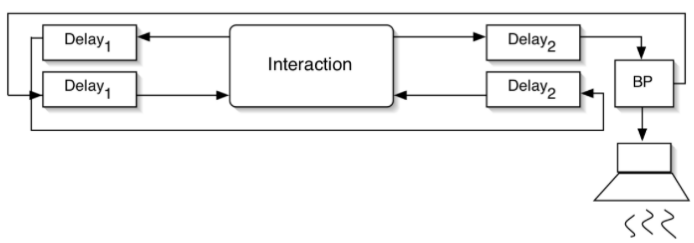


Figure 2: Circular banded waveguide, representing one beating mode of the bowl. $\text{Delay}_1 \neq \text{Delay}_2$ to achieve beating for asymmetric bowls. Presented in [7].

Band limiting

The bandpass filters were implemented as the 2.order filter in eq. (2), as this has normalized peak gain for all tunings.

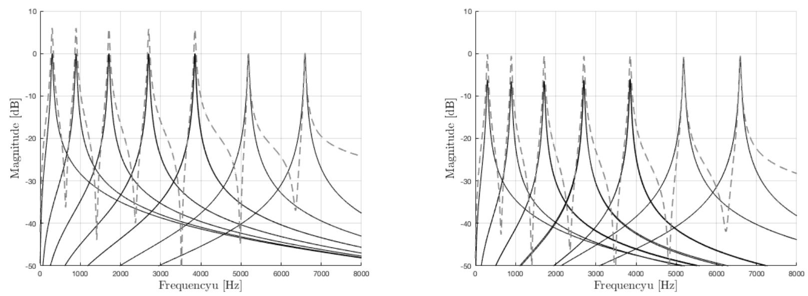
$$H(z) = \frac{1 - Rz^{\cdot 2}}{1 - (2R\cos(\theta))z^{-1}R^2z^{-2}} \tag{2}$$

The parameters of the transfer function has the following interpretation:

$$R = 1 - \frac{B}{2}$$

where B is the bandwith of the filter. This was chosen to sufficiently suppress neighbouring peaks in the comb response, as explained in[2].

$$\cos(\theta) = \frac{2R}{1+R^2}\cos(\psi)$$



(a) Unstable (b) Stable

Figure 3: Showing original and gain corrected responses. The dotted line is the summed response. The 5 lowest modes is beating, thus 2 bandpass filters with slightly different resonance frequency lies on top of each other.

where ψ is the normalized resonance frequency.

For a beating mode, the filters will be overlapping, and the summed response can exceed unity gain, leading to instability. By evaluating and summing the gain at ψ_k for all neighbouring filters, a gain correction $\tilde{A}_k$ was obtained. This corresponds to solving the system of N equations and N unknown:

$$\sum_{n=1}^{N} H_n(\psi_k) \cdot \tilde{A}_k = 1 \quad k = (1..N) \tag{3}$$

where N is the number of channels in the banded waveguide network.

Decoupling the modes
For real time applications where the bowl size potentially could be changed over time, the gain correction is a trivial procedure to carry out. Therefore another approach is taken, similar to that in STK. This is illustrated by the dotted lines in fig. 4. By decoupling the channels, the system is stable as long as all bandpass filters has below unity gain.

Allpass interpolation
The final implementation is illustrated in fig. 5. This also involves allpass interpolation filters, denoted AP, for implementing fractional delays. This was originally ignored, as frequency accuracy was found to be unimportant

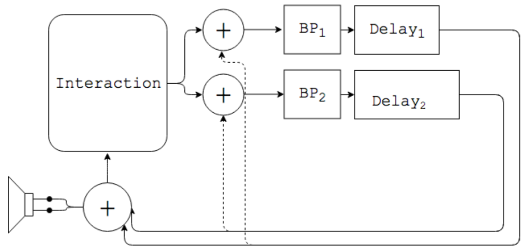


Figure 4: Block diagram of one beating mode. Notice 2 significant changes from fig. 2. 1. Beating modes are separated into 2 channels. Each channel has a direct feedback loop, i.e. no coupling.

for this particular instrument and its use. Though, for the matter of tuning the beating ratio accurately, fractional delays had to be introduced.

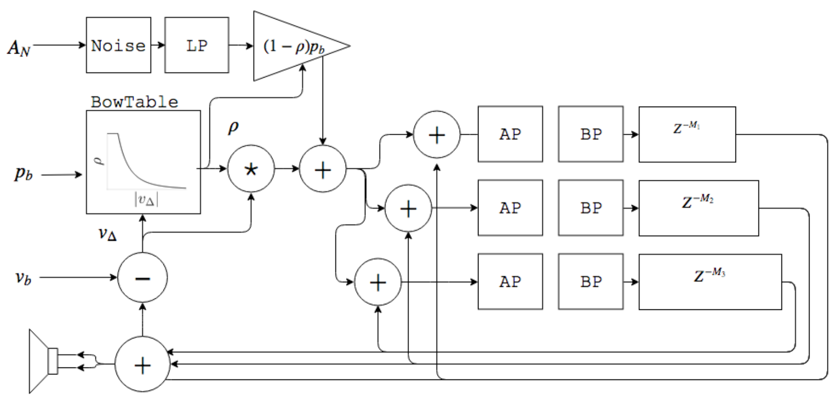


Figure 5: Blok diagram for the full implementation. BP = 2 pole bandpass filter. AP = 1 pole allpass interlation filter. Noise = Gaussian white noise, and LP = 2 pole butterworth lowpass filter with user controlled f_c. The actual number of channels is between 10 and 15 depending on f0, but this is left out for clarity

4.2. Excitation

The interaction between the puja and the bowl is modelled as a velocity dependent friction. The friction curve proposed in [2] was replaced by eq. (4), due to lack of clarity in the parameters it involved. Though, in the final implementation a bow table was used instead, this section describes the reasons behind this choice.

Nonlinear friction

The sustained excitation is nonlinear process, as a constant bow velocity results in sinusoidal output. When using a friction model for the interaction, oscillations arise due to a force threshold f_{max} being periodically overcome by forces in the string, resulting in a stick-slip motion.

First attempt in order to achieve stick-slip motion, was to implement the velocity dependent friction curve, eq. (4), and couple it with the system through the force relation in eq. (6)

$$\mu(v_\Delta) = \mu_d + (\mu_s - \mu_d)e^{-C|v_\Delta|}sgn(v_\Delta) \tag{4}$$

$$F_a = F_r \tag{5}$$

$$\mu(v_\Delta)v_\Delta = Z(v_\Delta^+ - v_\Delta) \tag{6}$$

Where Z is the wave impedance, $v_\Delta = v_b - (v^+ + v^-)$ is the sum of velocities at excitation position, and $v_\Delta^+ = v_b - v^+$ is the differential velocity at excitation position. As v_b and v^+ is known, respectively bow velocity and summed delay line outputs, the unknown is the outgoing velocity $v^- = v_\Delta - v_{\Delta^+}$, corresponding to the new input traveling into the bandpass filters.

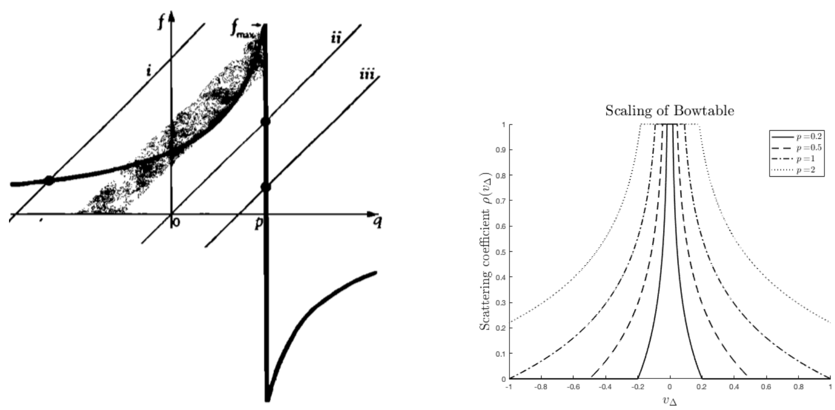
Figure 6 shows a graphical solution to the problem, where the linear segments I, II and III represents the reactive force at different time instances, and the nonlinear curve is the friction force. Their intersections are solutions to eq. (6). The infinitely steep part of the friction curve represents sticking. This model showed to be very sensitive to the (unknown) values of μ_d, μ_s, Z and C, and oscillating stick-slip motion was hard to obtain. But as solutions eq. (6) usually follow that same pattern, a bow table that simulate the behavior was used instead. This enforced the stik-slip motion, and reduced computation. It is implemented in STK as:

$$\rho(v_\Delta^+, p_b) = \min((|v_\Delta^+ \cdot (5 - 4 \cdot p_b)| + 0.75)^{-4}, 1) \quad [3] \tag{7}$$

Where p_b is bow pressure and ρ is a scattering junction coefficient that controls whether new input is added to the system, thus can be seen as an admittance for the puja on the bowl. The flat region $\rho = 1$ represents sticking and increasing p_b, increases this region. This is illustrated in fig. 6

With this approach to solving the coupling, the calculation of new input contribution v^- simplifies to a sum over `Nch`, a table lookup and a multiplication, as illustrated in the code-snippet below.

```
xk = sum(xkj);                      % Sum waves into interaction
vd = bowVelocity(k) - xk;           % Differential velocity
rho = bowTable(vd,bowPressure(k)); % Admittance
```



(a) (b)

Figure 6: (a) shows the Friedlander-Keller diagram for graphical solution in the force-velocity plane. [6]. (b) shows the bow table that approximates solutions to (a) when hysteresis is neglected. It relates admittance ρ to differential velocity, for varying bow pressure p_b. Notice how the region of sticking $\rho = 1$ increases with p_b

Excitation position

The excitation position was implemented with ”fixed” output pointers `oPtrs`, (incremented by the same value in each iteration), and lagging excitation pointers `ePtrs` which is modulatable. A pointer offset is calculated from an excitation position, $[0; 1]$ representing the entire circumference of the rim, scaled with the delay lengths, and added to `oPtrs` to obtain `ePtrs`

```
ePtrs = round(oPtrs+excPosition(k)*delays);
```

Adding noise

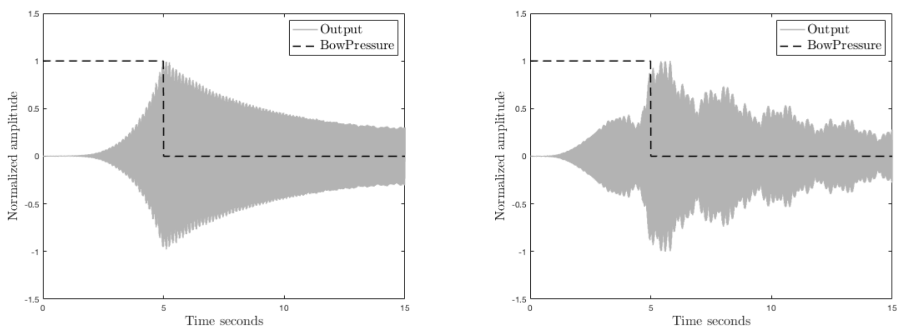
After successfully implementing the bowtable described above, sustained excitation with slow energy buildup was achieved, but the dry friction sound from puja-bowl interaction still wasn’t satisfying.

It was possible to enhance this friction sound by adding low-passed noise to model. The noise is coupled with the stick-slip motion of the puja by scaling with $1 - \rho$, such that noise is added during slipping. This was motivated by the physical behavior of friction, but remains a creative choice. Finally the noise is scaled by the bow pressure to ensure that no noise is added for $p_b = 0$.

Results are shown in fig. 8.

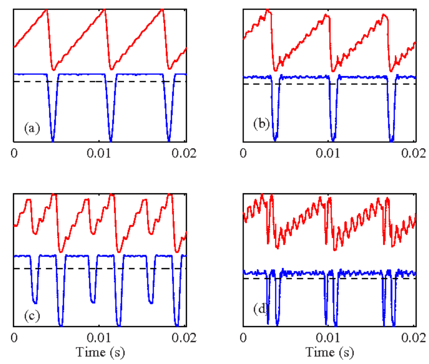
5. Results

This section illustrates the most important results. Explanations is found in the captions, and a discussion is given in the last section. Tests were made with sinusoidal excitation position, but only unrealistically high frequencies yielded audible results, and the implementation suffered from several artifacts. See section 7 for a discussion on this feature.

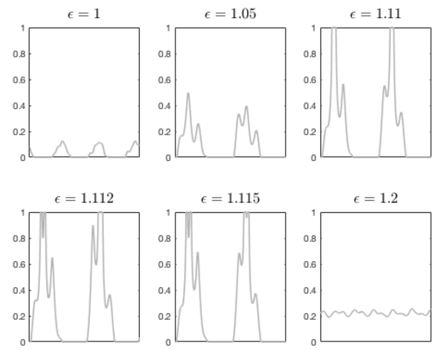


(a) (b)

Figure 7: Time domain plot of sustained excitation with $f_0 = 300$. **a)** has a beating ratio of 1.001 and noise gain of 0, where **b)** has beating ratio of 1.05 and noise gain of 0.1. Notice how the energy buildup is faster in **b)**, though they reach same energy level at the point where the bow is released (bowPressure = 0)

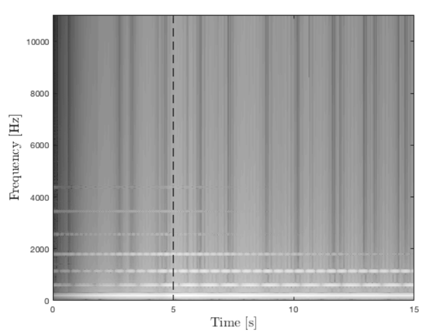


(a)

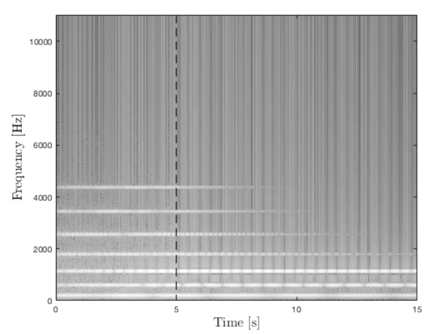


(b)

Figure 9: **a)** stick-slip helmholtz motion for bowed string in [6] Blue trace is string velocity at bowed point. **b)** Stick slip illustration for the synthetic bowl. $\rho = 1$ for sticking. ϵ denotes different bow pressures, and is a scalar applied to the lowest pressure that allowed frictional oscillation, i.e. $\epsilon = 1$. Notice that the stick-slip oscillation obtained in **b)** does not resemble any type of Helmholtz motion in **a)**. Notice also how the stick-slip pattern starts to disappear already for $\epsilon = 1.2$



(a) `noiseAmp` $= 0$



(b) `noiseAmp` $= 0.1$

Figure 8: Spectrogram of bowl at sustained excitation. The dotted line marks when the excitation is stopped. (a) shows how mainly mode 1 & 4 is excited with a noise free friction interaction, and (b) shows how all modes are excited when noise is present, and how the overall envelope has a shorter attack, as energy builds up faster

6. Conclusion

A model of the Tibetan bowl was successfully implemented in matlab, with banded waveguides and the STK bow table. The long decay was achieved by fine tuning the passband gains and decoupling the modes, which also enabled a flexible control of the beating modes. Using the frequency ratios obtained from [9] yielded realistic timbres, and the dry friction sound was successfully enhanced by adding noise to the model. This did though change the response of the system drastically, and in general the system showed to be very sensitive to the interaction parameters. Specifically self sustained excitation could only be achieved for a very small range of bow pressures, as illustrated in fig. 9, yielding very limited possibilities for interacting with the model. These issues are discussed in the next section.

7. Discussion & Improvements

Bowl size

As explained in 3.4 the size of the bowl is changed by simply scaling f0, according to one of 4 presets of frequency ratios. A smooth transition could be obtained by interpolating these presets, or by modelling the geometry and propagation speed of the bowl directly, and deriving the ratios from here.

Excitation position

For a realistic situation, the change in excitation position is very slow relative to the wave propagation, and therefore not audible[2]. Though, changing it extreme extremely fast gives an interesting vibrato-like effect, which could be used as a modulation tool. In order to remove the artifacts related to this feature, 2 improvements should be made:

- de-interpolation for continuously changing position
- two cross-fading read pointers to avoid colliding read-write pointers

Helmholtz motion

As shown in fig. 9, the oscillation at the friction interaction does not resemble any type of Helmholtz motion [6]. This is not surprising as Helmholtz motion is a concept describing the bowed string, where a harmonic comb response interacts with the friction model. To verify the results shown in fig. 9, further studies into stick-slip motion on inharmonic and beating resonators should

be made. Hopefully this could cast some light on the sensitivity towards bowing parameters, that currently limits the interaction possibilities.
Furthermore, the final implementation has no transition between bowing and striking, but instead 2 parts of the code that is conditionally evaluated based on the `doBow`. This is due to a limitation in the bow table: the width of its sticking region approaches 0 asymptotically as the pressure falls.
These results suggests that the bow table, as originally designed for bow-string interaction, is too crude a simplification in this application.
A possible improvement would be to introduce more physical accuracy the interaction scheme. For instance, a model that takes into account both radial and tangential velocity and pressure, as in [9], would describe both striking and bowing well, and need no explicit information on whether to bow or not. Furthermore it would allow changing material constants such as stiffness, mass and damping of the puja, to enhance the possibilities of interaction with the model.

Noise

When noise was added to the model, the sensitivity towards bowing parameters decreased drastically along with a drastic increase in output energy, resulting in a confusing user experience. Studying the effect of the noise on the system dynamics, could hopefully lead to a more stable implementation.

8. References

[1] K. Karplus A. Strong. Digital synthesis of plucked-string- and drum timbres. *Computer Music Journal*, 1983.

[2] P. R. Cook G. Essl, S. Serafin and J. O. Smith. Theory of banded waveguides. *Computer music journal*, 2004.

[3] S. Sinclair G. Scavone, M. Wenderley. Audio-haptic interaction with the digital waveguide bowed string. *CIRMMT*.

[4] J. O. Smith III. A new approach to digital reverberation using closed waveguide networks. *1985 International Computer Music Conference*, 1985.

[5] J. O. Smith III. Physical modeling using digital waveguides. *Computer music journal*, 1992.

[6] J. Woodhouse M.E. Mcintyre, R.T. Schumacher. On the oscillations of musical instruments. *Journal of the Acoustical Society of America*, 1983.

[7] Lus L. Henrique Octvio Incio. The dynamics of tibetan singing bowls. *International Symposium on Musical Acoustics (ISMA)*, 2004.

[8] J. O. Smith III S. A. Van Duyn. Physical modeling with the 2-d digital waveguide mesh. *1993 International Computer Music Conference*, 1994.

[9] J.O. Smith III S. Serafin, C. Wilkerson. Modeling bowl resonators using circular waveguide networks. *DAFx-02*, 2002.