Designing Mappings for the Sponge: Towards Spongistic Music

Martin Marier
Center for Interdisciplinary Research in Music Media and Technology (CIRMMT)
Faculté de musique
Université de Montréal
Montréal (QC), Canada
martin.marier.1@umontreal.ca

ABSTRACT

This paper reviews the evolution of the mapping strategies used for a cushion-like musical interface called the sponge. It describes how suggestions from the literature were concretely implemented and used in performance, offering a concrete example of a composer's approach to mapping. It also provides an example of a real world situation where no single strategy constituted a solution to the issue of mapping. It concludes that musical composition requires the use of a multitude of mapping strategies in parallel.

1. INTRODUCTION

The question of mapping for electronic musical interfaces has been researched and debated extensively in the literature. Many authors have proposed design guidelines[4, 8] and mapping techniques[6, 3, 1]. This paper reviews how these suggestions or recommendations were applied during the development process of a musical interface called the sponge, offering a concrete example of a composer's approach to mapping.

The sponge is a musical interface that resembles a cushion. Its development started in 2007 and is ongoing. The compositions, the mappings and the sponge itself were developed in parallel and, right from the beginning, some sponge music was performed live.

It is assumed that performing using a musical interface can be the core of a validation process of the mapping strategies used. How an audience responds and how a performer feels hardly constitutes quantifiable data, but it can be very revealing of how well the various aspects of a performance work... or not.

After briefly describing the objectives of the research and the musical interface itself, the numerous mapping strategies used in conjunction with the sponge will be reviewed.

2. GOALS AND ORIGINS

The sponge was designed for electroacoustic music performances. The vision was one of a performer standing on stage and performing a music that is normally associated with the acousmatic genre. The tradition in that field being to playback and spatialize prerecorded material on an orchestra of loudspeakers, the sponge would bring more humanity to the concert in allowing live interpretation.

The two main objectives that influenced design decisions during the sponge's development were (and still are):

1. To improve the interaction with an audience or other musicians;
2. To make the creation and performance of electroacoustic music more playful.

These two goals could only be attained by establishing a link between gesture and sound that felt as natural as possible. To achieve this, mapping guidelines proposed by various authors[4, 8] were followed and foam was chosen because it is a material that naturally requires effort to be bent or twisted.

3. DESCRIPTION

The sponge is approximately 19 cm × 5 cm × 28 cm in size. Its 14 sensors are connected to an Arduino-based microcontroller (the fio); the data is sent to a computer wirelessly using an XBee interface. Two force sensors detect when the sponge is squeezed; ten push buttons can be used for various purposes; and two three-dimension accelerometers sense its orientation at two points and can be used to extract many features such as bend, twist and shake. Both the mapping and the audio signal generation are implemented in the SuperCollider environment.

Figure 1: A picture of the sponge.

4. MAPPING STRATEGIES

In this section, the mapping strategies associated with each type of sensor are described.

Accelerometers

Sensing the deformations of the sponge always was at the center of the project. Force sensors are used to detect the squeeze, but they cannot detect twists or bends. For that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
NIME'14, June 30 – July 03, 2014, Goldsmiths, University of London, UK.
Copyright remains with the author(s).

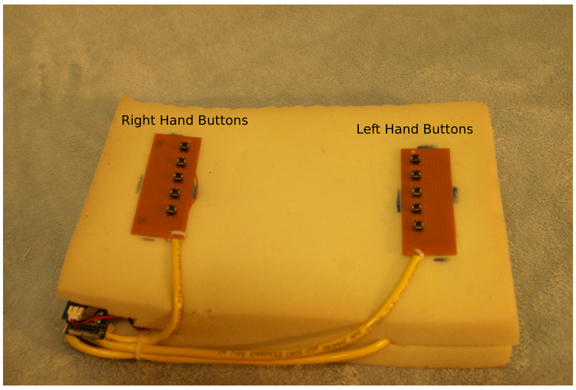

Figure 2: Underneath the sponge: buttons.

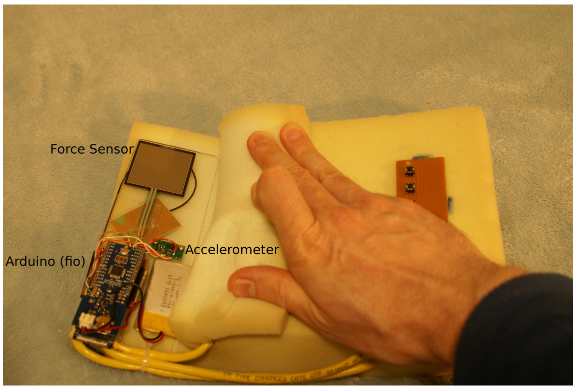

Figure 3: The right hand side, underneath the sponge. The Arduino board, a force sensor, an accelerometer, a battery and an antenna can be seen. The Xbee module is hidden under the Arduino board.

purpose, accelerometers were favored over bend sensors because their shape is not altered and are therefore less prone to breakage.[1] Another advantage is that they can also be used to sense orientation, shocks and shakes.

Twist, Bend and Other Features

Following the recommendations made by many researchers and well summarized by Hunt et al.[5], a multilayer mapping strategy was adopted. For the sponge, the first layer's role is to extract features such as *bend* and *twist*. These two are of particular interest because they constitute very *spongistic* gestures. Bend can be obtained by differentiating the pitch of the two accelerometers; twist can be obtained by differentiating the roll. If the sensor values are scaled between -1.0 and +1.0, bend can be calculated with:

$$bend = \arctan\left(\frac{x_2}{z_2}\right) - \arctan\left(\frac{x_1}{z_1}\right)$$

where x_1, z_1, x_2 and z_2 are the sensed accelerations in x and z of accelerometers 1 and 2. Replacing the accelerations in x by the accelerations in y will yield twist instead of bend.

It was observed that mapping the bend feature to the pitchbend of a synthesizer was very effective. This mapping is very engaging and, according to comments often made by audience members, it is naturally understood by most people. This probably has to do with the fact that this mapping evokes the behavior of physical objects such as the neck of a guitar or a bendable metal plate.

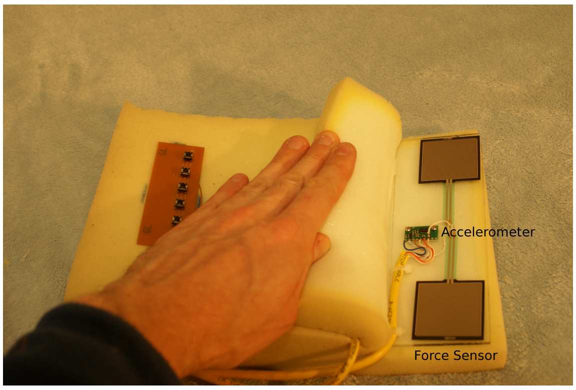

Figure 4: The left hand side, underneath the sponge. Two force sensors and an accelerometer can be seen. The bottom force sensor is disconnected; only two force sensors work on the device.

However, because of the limitations inherent to the use of accelerometers, this mapping does not work in all contexts. For example, if the sponge is held vertically, the acceleration in x will not be affected by the bend gesture, and the value of the bend feature will not change. Also, gestures such as wiggling, tapping or scraping will cause unwanted changes to the bend feature value. A good way to work around these two issues is to make that mapping active only while a button is pressed.

Experimental mappings in which *twist* was mapped to sound granulation parameters like granulation index or grain density were tried in the studio, but were never tried on stage. Such mappings were effective, but the issues that happened with the *bend* feature were also present and, in this context, the aforementioned workaround did not work as well. The timbre modulations were becoming too erratic when the sponge was held vertically, but deactivating this mapping using a button led to an unnatural feeling because the timbre was suddenly becoming too stable. It is interesting to note that this observation is in agreement with the recommendation that states that timbre should not be controlled directly [4]. Instead, an implicit mapping strategy was developed to control timbre.

Implicit Mapping

When mappings are explicit, their function's complexity increases with the complexity of the mapping (when designing many-to-many mappings, for example). It is generally admitted that complex mappings feel more natural and lead to more expressive instruments [4]. However, employing a strategy whose complexity will not grow indefinitely is probably a good idea.

Implicit mapping strategies are slightly more difficult to grasp for a user, but their complexity remains the same even if the mappings become more complex. Approaches that involve neural networks [6] or interpolation [3, 1] are probably the most common. The author chose to develop a preset interpolation system for the SuperCollider environment [7]. That strategy makes the creation of many-to-many mappings much easier.

Using this method, all the six axes of the accelerometers were mapped to more than 25 synthesis parameters (granulation and additive synthesis):

- Granulation index (× 5);

---

[1] During the last six years, none of the accelerometers used ever broke. On the other hand, many (more than ten) force sensors had to be replaced. It is suspected that bend sensors would have broken even more often.

- Grain duration (× 5);
- Delta time between each grain (× 5);
- Grain envelope attack time (× 5);
- Grain pitch ratio (× 5);
- Sine wave frequency;
- Sine wave amplitude.

The result is a very engaging mapping in which timbre is controlled holistically. The performer can stop thinking about parameters individually and can concentrate on musicality.

Filters: To Smooth or Not to Smooth

It can be tempting to smooth out sensor signals with low pass filters, especially during the development stage when it is very useful to visualize the data. However, high frequency control signal is not only noise, it also carries important gestural information that can turn out to be very expressive. The author succumbed to the temptation of filtering when he first attempted to design a mapping for the granular synthesizer. Sensor data was smoothed before being fed to the preset interpolation system, which ensured that all the granulation parameters would be moving smoothly. In turn, the granulation index and the delta time between grains would be randomized to avoid the metallic sounding artifacts that often appear in granulation processes. In the end, high frequency control data was removed at the mapping stage while, at the audio generation stage, high frequency noise was added again to get rid of the unnatural perfection of digital systems.

In a subsequent version, the smoothing filters and the noise generators were simply removed and the result was much more convincing:

- The latency inherent to smoothing filters was eliminated;
- There was less processing involved, making the whole algorithm less CPU intensive;
- The high frequency imperfections were not random anymore: they were related to gesture.

After that discovery, the importance of high frequency sensor data was no longer underestimated. In fact, the idea was pushed a step further and using a high pass filter on accelerometer data was tried. The resulting bipolar signal is very interesting because it can replace almost any random generator in any synthesis algorithm. Using it to excite a waveguide synthesizers is very effective because it ensures that the energy of the gesture is transferred directly to the energy of the sound. This is one of the most expressive mappings on the sponge and happens to be a simple one-to-one correspondence.

Force Sensors

The two force sensors on the sponge do very simple things. The first one is mapped directly to the frequency of a resonant low pass filter. This one-to-one mapping is certainly not the most expressive, but its simplicity makes it useful for the performer. When noisy granulated clarinet sounds go through that filter, modulating its frequency by squeezing the sponge can bring to mind the sound of wind or the waves, which has a strong musical significance.

The other force sensor is used as a damper for the waveguide synthesizer. In other words, pressing it shortens or mutes the resonance of an artificial string. It is interesting to note that Hunt and Wanderley4 recommend that ”energy should be required for amplitude”, while this mapping implements the opposite: energy is required to stop or to dampen the sound. This may seem contradictory, but both mappings model the behavior of acoustic instruments and both ideas are perfectly compatible. Mimicking the laws of physics will likely lead to mappings that feel natural.

Buttons

From the start, the author was reluctant to add buttons to the sponge. Such binary sensors do not have much expressive potential and they could hardly be helpful to establish a natural link between gesture and sound. It took four years of development before it was acknowledged that buttons could have their use. They were added in 2011 and are now an intrinsic part of the sponge. Because they are very useful to control macro processes, performing without them has become unthinkable. Here is a list of tasks that buttons can accomplish very well:

- Start/stop a synthesis or processing algorithm;
- Start/stop the recording of audio or control signals;
- Start/stop the playback of audio or control signals;
- Reroute audio or control signals;
- Activate/deactivate mappings;
- Step to a subsequent part in a sequence;
- Any combination of the above.

As a piece grows in complexity, the number of tasks to be accomplished by buttons can become very large. Depending on the interface used and on the musical needs, a one button per task strategy can become impractical. At a certain point while composing for the sponge, it became essential to develop a strategy that would allow the same buttons to be used for many different tasks. Such dynamic mappings have to be used with care because they can break the link between gesture and sound; but in this case, even though the aforementioned tasks can play an important musical role, they are not traditionally linked to any expressive gesture. The gesture of pushing a button is a generic one and does not convey any specific meaning; the risk of breaking a previously established convention is therefore very low.

Buttons to Control Pitch

Using buttons to control pitch is of particular interest because it has been employed by instrument makers for centuries. The valves of brasses, the buttons of accordions and the keys of woodwinds or keyboards are good examples.

At first, a one-button-per-pitch-class approach was tried with the sponge, but it was quickly dismissed because the range was too limited. Inspired by the functioning principles of the trumpet, a mapping that involved combining many buttons was developed: the binary values of five buttons were combined into an integer number that was mapped to a transposition offset (in semitones). This extremely simple idea turned out to have many advantages:

- When using five buttons, the range is two octaves and one fifth (32 semitones);
- Practicing scales or playing well-known melodies becomes possible and fun;

- A melodic virtuosity can be developed by a performer. Since the fingering is unique, the virtuosity that emerges is idiomatic to the sponge;
- Ornaments natural to human fingers are different from the ornaments played on any other instruments and are therefore *spongistic.*

Table 1: Using five buttons to control pitch. The binary numbers in the first column describes the fingering required to produce the pitch in the last column.

Buttons pushed	Transposition (semitones)	Pitch
00000	0	C1
00001	1	C#1
00010	2	D1
00011	3	D#1
00100	4	E1
...	...	...
11111	31	G3

A similar use of buttons is found on the Tooka instrument [2]. This kind of mapping contributes to the development of a musicality that is idiomatic to digital instruments.

Implementation

Controlling pitches or macro-processes with buttons makes perfect sense, but their layout needs to be clear, versatile and memorizable. In the current implementation, the behavior of each of the ten buttons is defined by a *mode*. It is possible to change to a different *mode* at any moment. Within a *mode* each button is either a *modifier* or an *executor*. An *executor* simply does something. The *modifiers* work like the *shift*, *control* or *alt* keys on computer keyboard: they change the behavior of *executors*.

An *executor* can do anything, including changing the mode. *Executors* have many levels. If no *modifiers* are held while an *executor* is pushed, it executes its level zero function. If a *modifier* is held, *executors* will execute the function that corresponds to that level.

5. TOWARDS MORE ARTICULATED SPONGE MUSIC

This paper reviewed the evolution of the sponge's mapping strategies over the last seven years. It described how suggestions from the literature were implemented and used in performance. The sponge is an example of a real world situation where no single strategy constituted a solution to the issue of mapping. Using a multitude of approaches proved to be more effective.

The interface is now mature: it is sturdy, the latency is hardly perceivable and the wireless link is reliable. While mapping strategies and hardware design will be developed further, the research focus will shift away from technical aspects to more artistic questions. Future research will center on practising, composing and defining what makes a music *spongistic*. Improvising with other musicians, especially other *spongists*, will help develop a deeper understanding of the interface and uncover its strengths and weaknesses. It is hoped that the mappings described combined with new strategies will eventually lead to the emergence of a musical language that is idiomatic to the sponge.

6. ACKNOWLEDGMENTS

The author would like thank Jean Piché, Garth Paine and Georges Forget for their insights, their guidance and their friendship. This research was financially supported by the FRQ-SC (*Fonds de recherche du Québec – Société et culture*), the CIRMMT (Centre for Interdisciplinary research in Music, Media and Technology) and the Université de Montréal.

The author is especially grateful for Annie Lalancette's continuous and invaluable support.

7. REFERENCES

[1] R. Bencina. The metasurface: applying natural neighbour interpolation to two-to-many mapping. In *Proceedings of the 2005 conference on New interfaces for musical expression*, pages 101–104. National University of Singapore, 2005.

[2] S. Fels and F. Vogt. Tooka: explorations of two person instruments. In *Proceedings of the 2002 Conference on New Interfaces for Musical Expression*, pages 1–6, Singapore, Singapore, 2002. National University of Singapore.

[3] C. Goudeseune. Interpolated mappings for musical instruments. *Organised Sound*, 7(02):85–96, 2003.

[4] A. Hunt and M. M. Wanderley. Mapping performer parameters to synthesis engines. *Organised Sound*, 7(02):97–108, 2002.

[5] A. Hunt, M. M. Wanderley, and M. Paradis. The importance of parameter mapping in electronic instrument design. In *NIME '02: Proceedings of the 2002 conference on New interfaces for musical expression*, pages 1–6, Singapore, Singapore, 2002. National University of Singapore.

[6] C. Kiefer. Exploring Timbre Spaces With Two Multiparametric Controllers. In *NIME '10: Proceedings of the 2010 conference on New interfaces for musical expression*, pages 3–9, Sydney, 2010.

[7] M. Marier. Designing Mappings for Musical Interfaces Using Preset Interpolation. In G. Essl, B. Gillespie, M. Gurevich, and S. O'Modhrain, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, Ann Arbor, Michigan, 2012. University of Michigan.

[8] G. Paine. Towards Unified Design Guidelines for New Interfaces for Musical Expression. *Organised Sound*, 14(02):142–155, 2009.