MUMT 620: Gestural Control of Sound Synthesis Project Report

Patrick Ignoto
Student I.D. 260280956

April 2016

Contents

1 Introduction **2**
1.1 Motivation 3

2 Background Information **4**
2.1 Previous work 4
2.1.1 Radio Baton 4
2.1.2 The MIDI Baton 4
2.1.3 Buchla Lightning 5
2.1.4 Digital Baton 5
2.1.5 Pinocchio 5
2.2 Gestural controllers, DMIs, and Mapping 6
2.2.1 Dimension space analysis of DMIs 7

3 Methods **8**
3.1 Max patch 8
3.2 Granul8 10
3.3 Mapping 12
3.3.1 One-to-one Mapping 12
3.3.2 One-to-many Mapping 14
3.3.3 Implicit Mapping 15

4 Discussion **17**
4.1 Dimension space analysis of The Batons 17
4.2 Mapping 19

5 Conclusion **20**

List of Figures

1.1 A look at the Batons as a DMI 2

2.1 A representation of a DMI by Miranda and Wanderley [1] . . . 6
2.2 The 7-axis Dimension space by Birnbaum et al. 7

3.1 Max Patch that sends sensor data to libmapper network . . . 9
3.2 Screenshot of the granul8 interface 10
3.3 Screen shot of the one-to-one mapping in webmapper GUI . . 12
3.4 Screenshot of the one-to-many mapping in webmapper GUI . . 14
3.5 Screen shot of Max/MSP patch developed to perform implict mapping 16

4.1 A dimension space analysis of The Batons 18

Abstract

The Batons are two wireless electronic devices that send input data to a distributed mapping network created with the libmapper software. This data can then be mapped to sound synthesis parameters of any synthesizer connected to this same distributed mapping network. The Batons can then be used to gesturally control this synthesizer and become a digital musical instrument (DMI). This report analyzes The Batons as a gestural controller and provides insight into certain mappings that were done to create a DMI with this controller.

Chapter 1

Introduction

A new form of gestural interface, called The Batons was developed as part of the final projects for the courses MUMT 619: Input Devices for Musical Expression and MUMT 620: Gestural Control of Sound Synthesis. This input data is transmitted to a base station connected to a laptop computer. The data is then taken in and made available on a distributed mapping network created using the libmapper software library. The data can then be mapped to controls in a synthesizer on the same distributed mapping network and the Batons can be used to gesturally control these sound synthesis parameters. This report will discuss the work done with libmapper and the reasoning behind the gestures and mappings done to make the gestural controller a true digital musical instrument (DMI) [1].

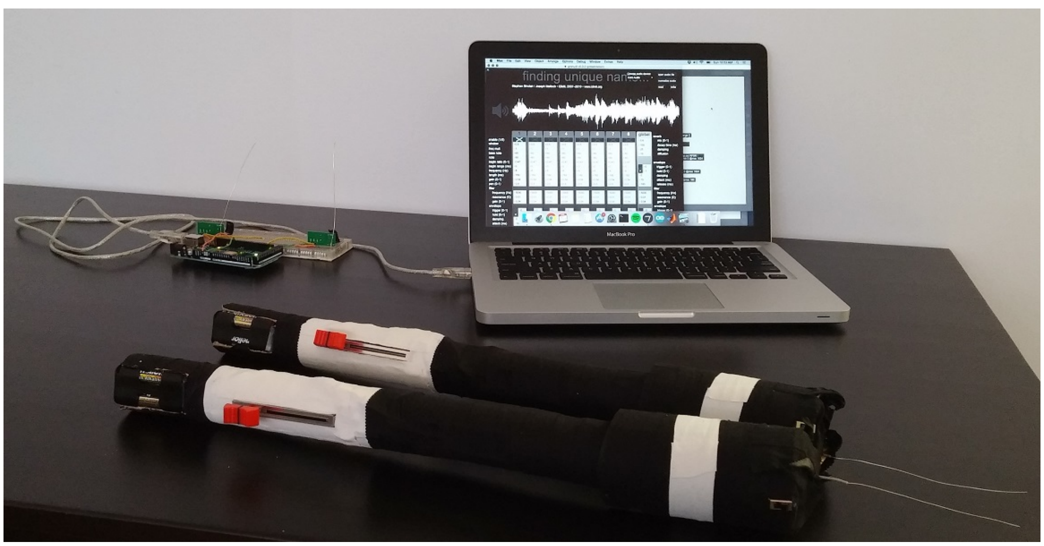

Figure 1.1: A look at the Batons as a DMI

Chapter 2 presents some previous work in baton-like gestural controllers and some background information on DMIs and mapping. Chapter 3 discusses the work done and the reasoning behind the mappings. The results of this work is then discussed in Chapter 4.

1.1 Motivation

The main motivation behind this project is to analyze and critique the gestural controller. It is also necessary to perform mappings of the various input signals to synthesizer parameters. This is to turn the gestural controller in to a true DMI using libmapper and the granul8 synthesizer. The hope is to use the materials learned in the course to create a DMI that attempts to reach the ideal of "low entry fee with no ceiling on virtuosity" [2].

Chapter 2

Background Information

The following chapter presents some background information regarding previous work done in baton-like devices and some background information on gestural controllers, DMIs, and mappings.

2.1 Previous work

The following section presents some background information on baton-like devices that have been created by other researchers. The devices are presented in chronological order of development.

2.1.1 Radio Baton

One of the earliest attempts to use batons for a DMI that uses conductor's gestures was the Radio Baton, a device created by Max Matthews that senses the conductor's gestures to control music [3]. The device used radio antennas to sense the batons held by the performer in three dimensions; the signal at the receiving antenna is dependant on how far the transmitting antenna (located on the baton) is [3]. The Radio Baton is then used to control a pre-programed score [3].

2.1.2 The MIDI Baton

The MIDI Baton is a device developed by researchers at Queen's University in Kingston in 1989. It consisted of a brass tube with a suspended conductive ball, that acted as a sort of accelerometer, because when the baton was accelerated by a user, it would close an electrical circuit and you could tell the direction the baton was accelerated [4]. With this you could then control a MIDI sequencer in real-time performance, as well as coordinate live performers at the same time [5].

2.1.3 Buchla Lightning

The Buchla Lightning is an instrument initially created by Don Buchla. The device uses IR LED tracking of two batons. Music is created by assigning MIDI events to various hand gestures [6]. For example, in some videos of performances with the Buchla Lightening, different drum sounds would be played when gesturing in different directions with the held batons. As one review of the device says, "You can bang on an invisible drum set, play air-marimba, conduct a phantom orchestra, or spew forth a dizzying cacophony of avant-garde noise" [6].

2.1.4 Digital Baton

The Digital Baton is a baton-like DMI created by Teresa Marrin-Nakra that attempts to imitate conductor gestures as closely as possible [7]. It is a single baton that uses a 3-axis accelerometer, force sensing resistors, and IR LED tracking as sensors. This data is then used, for example, to "trigger and shape multiple layers of sound in a live, interactive show" [8]. The device had a few drawbacks: it was uncomfortable to hold, required exaggerated gestures to achieve what the performer wanted, and it was difficult for the audience to determine how the baton was controlling the sound [8].

2.1.5 Pinocchio

A more recent use of batons for DMIs is the Pinocchio system developed by researches at the Technical University of Munich. This device's unique feature is that any baton device, including an actual conductor's baton, can be used to conduct a virtual symphony [9]. The device tracks the baton with a camera setup and image processing algorithms; an algorithm then conducts a virtual orchestra based on the gestures of the performer [9].

2.2 Gestural controllers, DMIs, and Mapping

The following section provides some background information on gestural control that is required for the rest of this report. Gestural controllers can be seen as the devices that provide human input through gestures to a DMI [1]. These inputs can then be mapped to sound synthesis parameters either independently or through a mapping software such as libmapper [1].

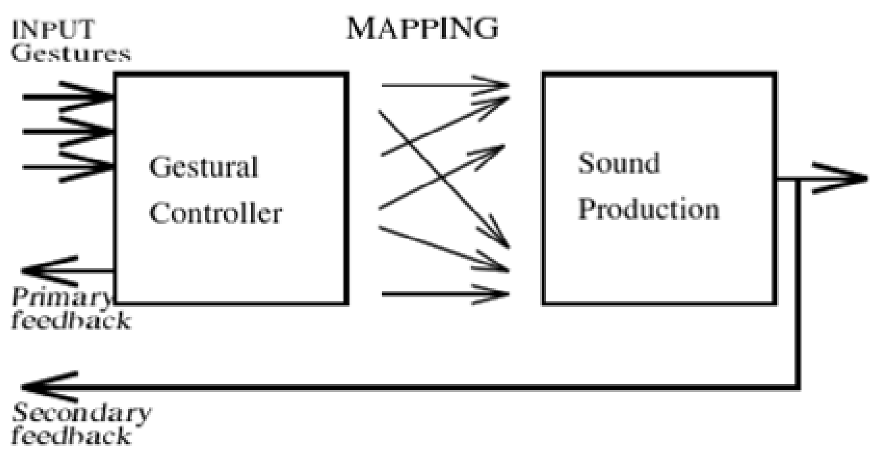

Figure 2.1: A representation of a DMI by Miranda and Wanderley [1]

The concept of a DMI depicted in Figure 2.1 separates the gestural controller and sound generator, therefore a mapping can combine the two in any desired way [1]. There are two strategies to perform mappings: implicit and explicit mapping [1]. Implicit mapping uses tools such as interpolation, feature extraction, or pattern recognition to perform a mapping between the various gestural input to the sound synthesis parameters. Explicit mapping is where the user explicitly defines the mapping between gestural input and sound synthesis parameters. This can be in a *one-to-one* manner, where one and only one input controls one sound parameter, in a *one-to-many* manner, where one input controls many different sound parameters, and in a *many-to-one* manner, where a combination of many inputs control one sound synthesis parameter.

2.2.1 Dimension space analysis of DMIs

Birnbaum et al. proposed a way to visualize the different qualities of a DMI [10]. The technique is derived from dimension space analysis and plots seven different qualities of a DMI in a 7-axis dimension space [10].

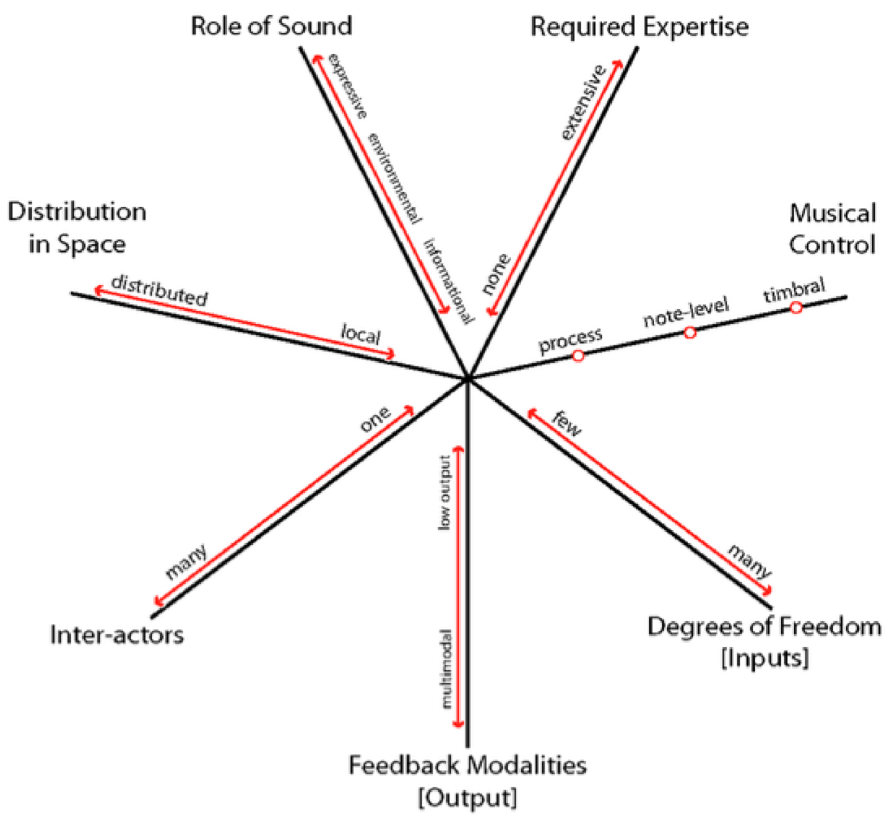

Figure 2.2: The 7-axis Dimension space by Birnbaum et al.

The 7-axis dimension space plots several different dimensions on the plot shown in Figure 2.2. For example, *Degrees of Freedom* plots the amount of input signals the controller has. This method was used to analyze the Batons as a DMI in Chapter 4.

Chapter 3

Methods

The following chapter describes the work involved in getting the gestural controller to work with libmapper, and the mappings that were generated with the granul8 synthesizer by Stephen Sinclair and Joseph Malloch.

3.1 Max patch

The Batons send their data via the serial port of an Arduino board connected to the computer. The "Receiver station" takes the transmitted serial data from each Baton and combines them in the following way: left baton pitch, left baton roll, left baton slide position sensor value, left baton FSR value, right baton pitch, right baton roll, right baton slide position sensor value, right baton FSR value. This packet of data is then transmitted via the main serial port to the MacBook Pro connected to this receiver station via USB. A patch was then created in Max/MSP to read this serial data coming from the receiver station and send that out to the distributed mapping network created by libmapper.

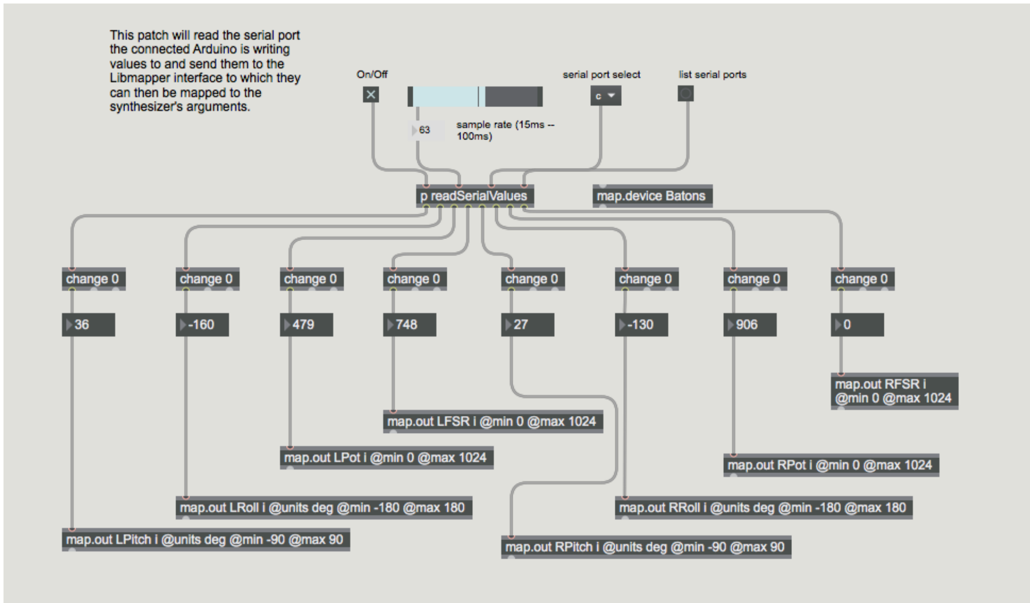

Figure 3.1: Max Patch that sends sensor data to libmapper network

The Max patch pictured above will read the serial port, convert the values into integer numbers, and, if there is a change in the last time they were transmitted, the values are sent out to the distributed mapping network created with libmapper. Checking for a change from the last time they were transmitted was done to limit the amount of data being sent to the synthesizer. Sending all 8 pieces of data via libmapper to the granul8 synthesizer from both batons at the rate of transmission seemed to cause Max/MSP to crash. Only sending the data on a change reduced the frequency of Max/MSP crashing.

3.2 Granul8

Granul8 is a granular synthesizer developed by Stephen Sinclair and Joesph Malloch for the McGill Digital Orchestra project [11]. It allows control for eight independent grains and is meant to be compatible with libmapper [11].

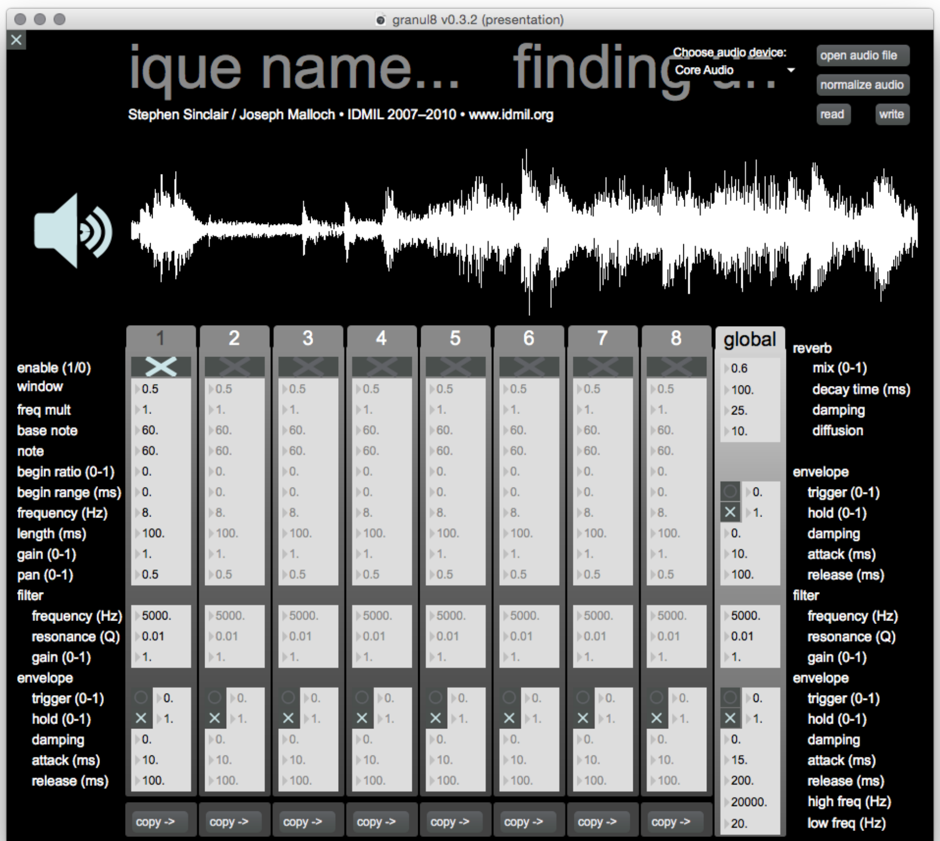

Figure 3.2: Screenshot of the granul8 interface

However, the granul8 synthesizer uses an older libmapper "mapper" object for Max/MSP. With the version of libmapper installed on the computer used for this project, this "mapper" object was not working properly. The device would show up in the webmapper GUI, but signals did not. So the synthesizer was modified as part of the project and the "Experiences with mapping" assignment. The new libmapper objects for Max/MSP were used and signals were added for each of the various controls over a grain. Because of the structure of the patch, however, the control over 8 grains was lost when using signals being sent via libmapper. Originally, the various signals for each of the 8 grains was defined with a JSON object that was included with the synthesizer. This was lost when adding a "map.in" object to the various controls for the grain. Despite this, one grain can be controlled via libmapper, and this was sufficient to perform a simple granular synthesis gesturally.

3.3 Mapping

3.3.1 One-to-one Mapping

As a test of the capabilities of the Batons, a simple one-to-one mapping was done via the libmapper GUI. Through trial and error, a comfortable mapping that produced a good quality sound using all input signals except for the roll was developed.

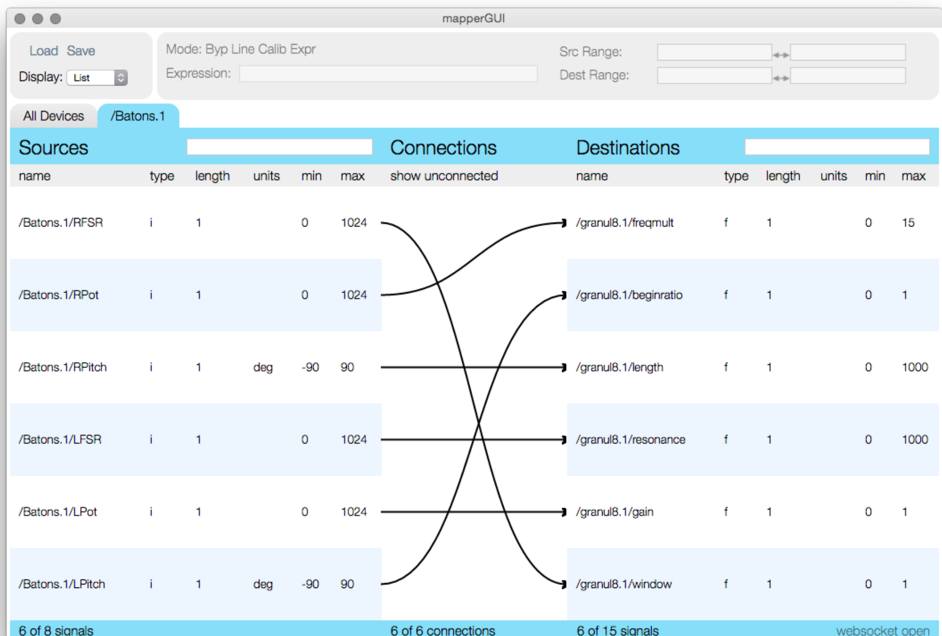

Figure 3.3: Screen shot of the one-to-one mapping in webmapper GUI

The pitch of the left hand scrubs the "begin ratio" control, which determines where the granular synthesizer begins taking grains. This is much like the scrubbing motion done with the mouse in the synthesizer's interface. The pitch of the right hand controls the length of the window that grains are taken from. This is again controlled with a scrubbing motion in the interface. Therefore, this was a comfortable way to control these particular setting and allows expressivity. The pitch of the two batons control two settings that have a very direct and noticeable control over the sound, providing easily understood auditory feedback to the player.

In a similar way, the choices made for the mapping of the linear position sensors were done to take advantage of the gesture. The linear position sensors are not typically used as a continuous control like the pitch of the batons. Frequency multiplier and gain are two settings of the granul8 synthesizer that do not need a continuous amount of control when used. Those two settings are set once and then adjusted when the timbre or loudness needs to be modified. Therefore mapping these two settings to the linear position sensors provided a type of gestural control necessary to effectively use these two settings in the synthesizer. The right linear position sensor was mapped to the frequency multiplier setting and the left linear position sensor was mapped to the gain.

Finally, the FSRs were mapped to settings that could remain a base value, and only needed to be modified when timbral changes were necessary. This was mapped to the window shape and resonance settings of the synthesizer. When the performer believed the timbre needed to be changed, the FSR could be pressed on to modify these settings temporarily, which could produce some interesting timbral modifications.

3.3.2 One-to-many Mapping

In an attempt to produce more interesting timbres with the Batons, a one-to-many mapping was done via the libmapper GUI. Hunt and Kirk claim that "mappings that are not one-to-one are more engaging to users" [12]. As well this mapping was used to test to see if there was a difference when controlling various synthesizer settings with the left or right hand. Again through more trial-and-error, a mapping that produced interesting timbres with the Batons and granul8 synthesizer were produced.

Figure 3.4: Screenshot of the one-to-many mapping in webmapper GUI

The pitch was mapped to the synthesizer settings that benefit more from scrubbing motions. In this mapping the begin ratio was mapped to the right pitch to see if there was a difference in scrubbing with the right or left baton. It is the only setting mapped to the pitch of the right hand as the begin ratio is one of the more important settings and fine control is necessary. The pitch of the left hand is mapped to both the length and gain synthesizer settings, which allows a more interesting dynamic control over the sounds being synthesized. It also gives more of a conductor like feel to the Batons, as the right hand controls a time-related setting and the left hand controls dynamics.

Similar to the previous mapping, the frequency multiplier is controlled with one of the linear position sensors. To see if there was a difference in control with this motion, the hand that controls this setting was switched. It is now controlled with the left hand. The right hand controls the resonance and the frequency. A finer control over the resonance provided much more interesting timbres. As well, the begin range was controlled to slightly change the sound output when changing the resonance, which provided intriguing timbres.

Finally, only the right FSR was used to control synthesizer settings. It controls the frequency of the grains which changed the timbres in a compelling way that was not used in the previous mapping. It also controls the window shape setting to provide some variance to the output sound.

3.3.3 Implicit Mapping

An attempt was made to implicitly map the pitch and roll of each baton to three synthesizer settings in granul8, using the mnm library for Max/MSP by IRCAM [13]. This was done to attempt to use both pitch and roll to control the begin ratio, gain, and length settings of granul8 and to produce a much more interesting mapping with these input signals. The following Max/MSP patch was developed for this:

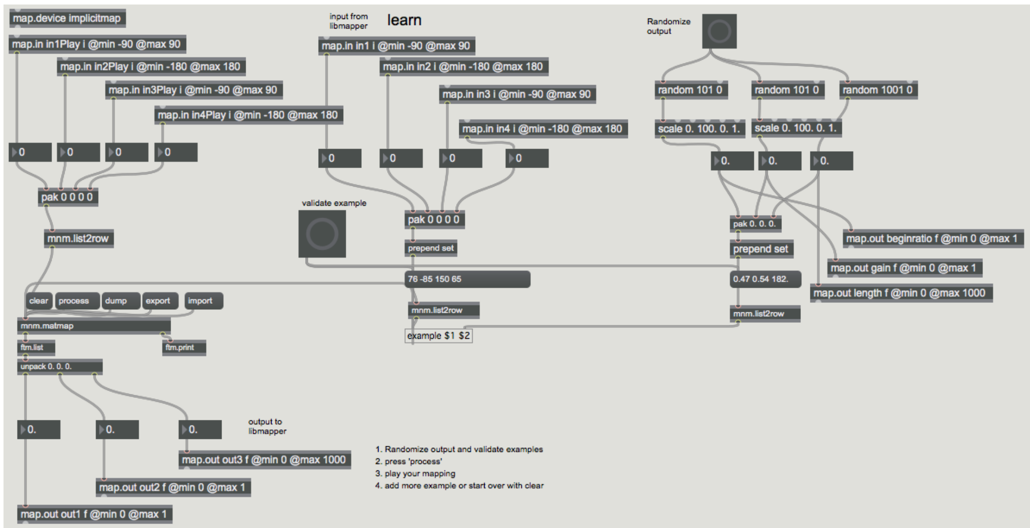

Figure 3.5: Screen shot of Max/MSP patch developed to perform implict mapping

The patch randomizes output values for the synthesizer and sends them to the synthesizer via libmapper. When transmitting data from the Batons, it takes in the baton orientations a performer would want to use for this randomized sound and this can be used to train the machine learning algorithm of the mnm library [13]. Several attempts were made but a workable implicit mapping could not be generated with these tools. The range of the outputs continued to fall outside of what was possible with the synthesizer and therefore crashed the synthesizer. This is likely caused by the pitch and roll having a very large range of possible values, much less than what is possible with the synthesizer, which caused the algorithm to keep suggesting values outside of the range it was taught. Large discontinuities in the roll values could also be a culprit; see Section 4.2 for more details.

Chapter 4

Discussion

The following chapter provides some discussion on the Batons as a DMI and the mappings that were generated. Overall, the Batons work quite well as a gestural controller, all the degrees of freedom of the devices were accessible and could be changed comfortably with either hand. The mappings provided a good look into how the Batons performed as a gestural controller. They are surprisingly light because of the materials used. The heaviest part is the battery pack, which was placed where the user would grip the batons, providing a very good balance to the Batons. Gesturing with the Batons is quite simple and someone with minimal knowledge could pick it up and start playing with it, provided the software is setup and configured. The Batons, using the granul8 synthesizer as a sound generator, provide a rather intriguing form of DMI. Further refinements and improvements to the mappings could make a very expressive and excellent DMI.

4.1 Dimension space analysis of The Batons

The Batons were analyzed using the dimension space analysis proposed by Birnbaum et al. [10].

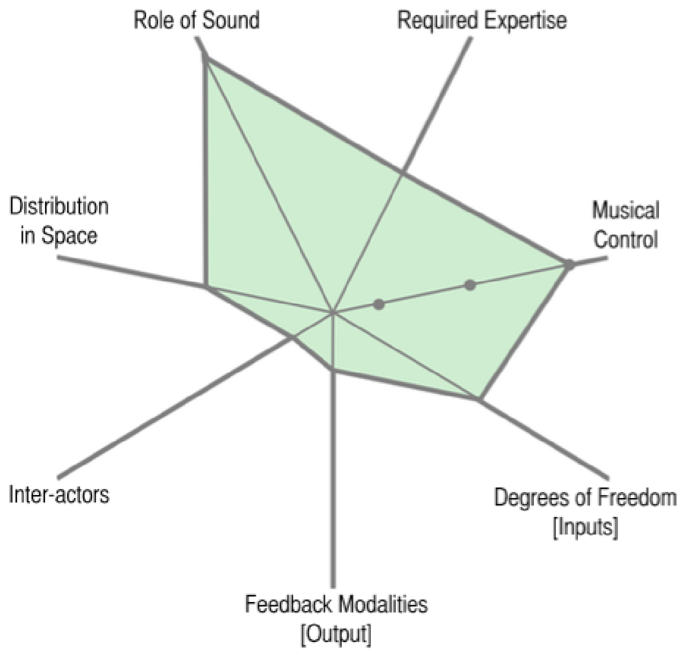

Figure 4.1: A dimension space analysis of The Batons

The *Role of sound* for the Batons is expressive as it is done for artistic purposes. The *Required Expertise* is an intermediate level as any one can play it and the sound will change based on what their movements. However, a good knowledge of how the synthesizer works with the DMI will produce the best results. The *Musical control* is on the timbral-level as timbres are being adjusted when using The Batons. There are 8 *Degrees of Freedom*, which are the number of input signals transmitted by the Batons. Only auditory feedback is available from the Batons, and consequently, *Feedback modalities* is close to the centre. There can only be one performer so the *Inter-actors* is also close to the centre. Since The Batons operate using libmapper, the *Distribution in Space* is across the same local area network, so it was placed a little further from the centre.

Overall, the dimension space analysis pictured in Figure 4.1 is rather unique when compared to some of the DMIs pictured in the paper by Birnbaum et al. [10], but is likely very similar to the DMIs described in Section 2.1.

4.2 Mapping

Only a few interesting gestures could be mapped with the Batons, as only explicit mappings were feasible. The roll was especially difficult to explicitly map because of its range. If it was slightly turned clockwise, it would be in the range of 170° - 180°. If slightly turned counterclockwise, it would be in the (-170°) - (-180°) range. This made mapping this parameter quite difficult as it caused large differences in sound with only small imperceptible gestures. Perhaps in future iterations of the Batons, some scaling factor should be applied in the firmware to get the roll in a more natural range. These large discontinuities in the roll are likely why the implicit mapping patch could not compute a useful mapping in the correct range.

The one-to-many mapping did provide more engaging mappings then the simpler one-to-one mapping. During a demo of these mappings, the small audience found that the one-to-many mapping was slightly more engaging. However, it was more noticeable to them what each degree of freedom was controlling in the one-to-one mapping. The audience could tell that the one-to-many mapping was controlling the sound through gestural movements, but could not specify how this was happening. To be more engaging to an audience, they should be able to determine what the gestures relate to.

Unlike many of the DMIs described in section 2.1, the gestures were far from being similar to a conductor's gestures. The mappings only took into account the pitch of the baton which left much to be desired. Some better thought out mappings with a different synthesizer could however mimic a conductor's gestures quite easily.

Chapter 5

Conclusion

The Batons are a gestural controller that were made into a DMI by mapping its input signals to sound synthesis parameters in a synthesizer. A Max patch took the values of pitch, roll, linear position sensor value, and FSR value for each baton and sent it to a distributed mapping network using the libmapper software. These were then mapped to sound synthesis parameters in the granul8 synthesizer that is on the same distributed mapping network. Two explicit mappings were then done: a one-to-one mapping and a one-to-many mapping. The one-to-many mapping produced more interesting timbres than the one-to-one mapping confirming Hunt and Kirk's claim [12]. An attempt to do implicit mapping using the mnm library by IRCAM was done, however, the output of this kept being out of the possible range and would crash the synthesizer. This is likely due to large discontinuities in the roll input signals or a range of values that is not easily processed by the mnm library. However, the use of the Batons as a DMI was overall a success and more time and effort in mappings would likely produce a very intriguing and expressive form of DMI.

Bibliography

[1] E. Miranda and M. Wanderley. *New Digital Instruments: Control and Interaction Beyond the Keyboard*. A-R Publications, Middleton, Wisconsin, 2006.

[2] D. Wessel and M. Wright. Problems and prospects for intimate control of computers. *Computer Music Journal*, 26:3:11–22, Fall 2002.

[3] T. Hong Park. An interview with max mathews. *Computer Music Journal*, 33(3):9–22, 2009.

[4] D. Keane and P. Gross. The MIDI Baton. In *Proc. of the 1989 International Computer Music Conference. San Francisco, Calif.: International Computer Music Association.*, pages 151–154, 1989.

[5] D. Keane and K. Wood. The MIDI Baton III. In *Proc. of the 1991 International Computer Music Conference. San Francisco, Calif.: International Computer Music Association.*, 1991.

[6] R. Rich. Buchla lightning ii. *Electronic Musician*, 12(8):118, 1996.

[7] T. Marrin and J. Paradiso. The Digital Baton: A Versatile Performance Instrument. In *Proc. of the 1997 International Computer Music Conference. San Francisco, Calif.: International Computer Music Association.*, pages 313–316, 1997.

[8] T. Marrin-Nakra. *Trends in Gestural Control of Music*, chapter Searching for Meaning in Gestural Data: Interpretive Feature Extraction and Signal Processing for Affective and Expressive Content. Ircam - Centre Pompidou, 2000.

[9] B. Bruegge et al. Pinocchio: Conducting a virtual symphony orchestra. In *Proceedings of the International Conference on Advances in Computer Entertainment Technology*, ACE '07, pages 294–295, New York, NY, USA, 2007. ACM.

[10] D. Birnbaum et al. Towards a dimension space for musical devices. In *Proceedings of the 2005 Conference on New Interfaces for Musical Expression (NIME-05)*, pages 192–195, New York, USA, 2005.

[11] J. Malloch. Granul8 synth. Available: `https://josephmalloch.wordpress.com/portfolio/granul8/`, 2010.

[12] A. Hunt and R. Kirk. Mapping strategies for musical performance. In M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*. IRCAM - Centre Pompidou, Paris, 2000.

[13] F. Bevilacqua et al. MnM: a Max/MSP mapping toolbox. In *Proceedings of the conference on New Interfaces for Musical Expression*, pages 85–88, Vancouver, Canada, 2005.